SECURE LIGHTWEIGHT AUTHENTICATION FOR INTERNET OF THINGS

M.Tech Thesis

Submitted in partial fulfillment of
the requirements for the award of the degree
of
Master of Technology
in
Software Engineering
submitted by

Harshit Tyagi (23/SWE/08)

under the guidance of

Dr. Divyashikha Sethia
Associate Professor
Department of Software Engineering

DEPTT. OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY, DELHI

June 2025

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that M.Tech Thesis entitled **Secure Lightweight Authentication for Internet of Things** which is submitted by Harshit Tyagi, Roll No - 23/SWE/08, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree Master Of Technology (Software Engineering) is a record of the candidate work carried out by him under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: 19 May 2025

May

Place: New Delhi

Dr. Divyashikha Sethia
Associate Professor
Department of Software Engineering
Delhi Technological University

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I Harshit Tyagi (23/SWE/08) hereby declare that the work which is being presented in the thesis entitled **Secure Lightweight Authentication for Internet of Things** in partial fulfilment of the requirements for the award of the degree of Master Of Technology submitted in the Department of Software Engineering, Delhi Technological University is a bonafide record of my own work carried out during the period from August 2023 to June 2025 under the supervision of Dr. Divyashikha Sethia.

The material contained in the thesis has not been submitted by me for the award of any other degree of this or any other institute.

Date: 20 May 2025

Place: New Delhi

Tyagi

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

May 21 2025

Signature of Supervisor

24.6.2025

Signature of External Examiner

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I am very thankful to **Dr Divyashikha Sethia** (Associate Professor, Department of Software Engineering) and all the Department of Software Engineering faculty members at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing me with the laboratories, infrastructure, testing facilities, and environment, which allowed me to work without obstructions.

I would also like to express my appreciation for the support provided to me by the lab assistants, seniors, and our peer group, who aided me with all the knowledge they had regarding various topics.

Harshit Tyagi

(23/SWE/08)

ABSTRACT

The proliferation of Internet of Things (IoT) devices across various domains has introduced new challenges in ensuring secure and efficient communication over inherently insecure networks. Authentication protocols in such environments must balance robustness, lightweight execution, and resilience against evolving attack vectors. Given the limitations of conventional schemes in resource-constrained and high-risk settings, this thesis report presents two novel contributions designed to enhance authentication security in IoT ecosystems through cryptographic and architectural innovations.

As part of this effort, the first contribution targets security enhancement in Internet-of-Medical-Things (IoMT) scenarios. Robust schemes are particularly critical in such settings due to the transmitted data's sensitivity and resource-constrained device limitations. While Masud et al. proposed a protocol for securing data in IoMT networks, their approach remains vulnerable to offline password-guessing and privileged insider attacks, posing serious privacy and patient safety risks. To address these issues, this report proposes a novel protocol, *P-MASFEP* (security-enhanced PUF (Physically Unclonable Functions)-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI (Public Key Infrastructure)). P-MASFEP integrates PUFs with fuzzy extractors to actively derive stable cryptographic keys from biometric input, mitigating password-guessing risks. It also employs PKI to distribute session keys securely and ensures protection against insider threats through mutual authentication.

The second contribution focuses on overcoming the inherent limitations of a traditional authentication framework, Kerberos. Its traditional design faces challenges in resource-constrained IoT environments, including computational inefficiencies, lack of clock synchronization, and limited scalability. In addition to these limitations, Kerberos remains vulnerable to several modern attacks such as password-guessing, Kerberoasting, Golden Ticket, and Silver Ticket attacks. Prapty et al.'s *KESIC*, adapts Kerberos for IoT by introducing optimizations. However, it relies on symmetric cryptography for authentication and key exchange. Additionally, it remains susceptible to password-based attacks, necessitating a more secure approach. This work proposes two novel protocols to address these issues: (1) *Kerberos with FIDO (Fast Identity Online) Integration (KFI)*, which integrates FIDO's passwordless authentication to eliminate password-derived vulnerabilities; and (2) *Kerberos with FIDO and Lightweight extension for IoT (KFLIT)*, which extends KFI by incorporating lightweight HMAC and XOR operations to reduce computational overhead, counter-based synchronization to eliminate dependency on real-time clocks, and an attestation mechanism to verify IoT device integrity before granting access.

Together, the proposed solutions address critical gaps in current authentication mechanisms for constrained environments. By tackling domain-specific (IoMT) and general-purpose (IoT) challenges, this report contributes to building a secure and scalable authentication foundation for next-generation connected systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Internet of Things (IoT) has transformed how devices interact and exchange data across domains such as healthcare, industrial automation, and smart cities. This paradigm shift enables seamless, real-time communication between heterogeneous devices, offering automation, efficiency, and enhanced decision-making capabilities. One of the most impactful verticals emerging from this revolution is the Internet of Medical Things (IoMT), which integrates IoT technology into medical systems to facilitate remote monitoring, diagnosis, and treatment [1, 2].

However, this hyperconnectivity brings forth substantial security and privacy concerns. IoT devices operate in resource-constrained environments with limited computational power, memory, and battery life. Furthermore, they are frequently exposed to open or semi-trusted networks, making them vulnerable to a wide range of attacks [3, 4]. In a sensitive domain such as IoMT, these vulnerabilities become particularly alarming as they can lead to unauthorized access to patient records, manipulation of medical data, and even disruption of critical services, potentially endangering patient lives [5–7].

In response, the research community has proposed several lightweight authentication protocols to mitigate these threats while maintaining efficiency in constrained environments. Many schemes leverage cryptographic primitives such as Physically Unclonable Functions (PUFs), biometric-based key generation, and optimized handshake procedures to ensure secure access with minimal overhead [8–10]. One notable contribution is Masud et al.'s Mutual Authentication and Secret Key (MASK) protocol [11], which combines hash functions, nonces, and XOR operations for mutual authentication in IoMT. While MASK achieves efficiency and resistance to many known attacks, it remains susceptible to offline password guessing and insider threats.

Simultaneously, efforts have been made to adapt well-established and robust protocols like Kerberos to meet the security demands of modern IoT environments. Initially designed for enterprise networks, Kerberos offers strong mutual authentication through a ticket-based mechanism [12, 13]. However, its reliance on synchronized timestamps, computationally intensive encryption, and password-based logins makes it unsuitable for IoT ecosystems, where real-time clocks and resource availability are limited [14]. To overcome these limitations, researchers have proposed enhancements such as integrating biometric credentials [15], adopting public key cryptography [16], and employing HMAC-based cryptography with counter synchronization [17]. While these modifications improve efficiency, many still inherit Kerberos' susceptibility to password-based attacks, including Kerberoasting, Silver Ticket, and Golden Ticket exploits [18–20].

Against this backdrop, the need for authentication schemes that are both lightweight and resilient against modern threats has become paramount. This thesis explores two such advanced protocols: one that enhances lightweight mutual authentication using fuzzy extractors and PKI for secure IoMT deployments and another that reimagines Kerberos with FIDO-based passwordless authentication and IoT-focused optimizations. Together, these solutions aim to contribute toward building secure, scalable, and practical authentication mechanisms for the evolving landscape of interconnected devices.

1.1 Background

Several authentication protocols have been proposed to address the security concerns in IoT environments. Alladi et al. [9] proposed a low-power Healthcare Authentication protocol using resource-constrained IoT devices (HARCI). The protocol leverages PUFs to generate secure session keys for two-way authentication and ensures end-to-end security between network devices using distinct session keys. Ashraf et al. [8] highlighted the importance of secure remote user authentication to protect sensitive data in healthcare. Their protocol emphasizes lightweight cryptographic techniques to reduce computational overhead and transmission costs while maintaining authorized access. Gaba et al. [10] proposed a Lightweight Key Exchange (LKE) protocol for the fast-evolving Industrial Internet of Things (IIoT) using certificate renewal to ensure robust security. Gope et al. [21] designed a physically secure key establishment scheme for Industrial Wireless Sensor Networks (IWSNs), aiming to enhance the reliability and trustworthiness of network communications. Shao et al. [22] developed a secure PUF-based authentication protocol tailored for Wireless Medical Sensor Networks (WMSNs) to safeguard patient data. Masud et al. [11] introduced the Mutual Authentication and Secret Key (MASK) protocol, which utilizes lightweight cryptographic primitives such as one-way hash functions, nonces, PUFs, and bitwise XOR operations to provide secure communication in IoT-based healthcare.

To adapt widely used and well-established protocols such as Kerberos to the demands of IoT environments and modern security challenges, researchers have proposed various enhancements to reduce its computational and synchronization overhead. Downard [16] extended Kerberos using public-key cryptography (PKC), replacing password-based authentication with digital signatures to improve scalability. Han et al. [15] enhanced Kerberos for mobile computing by embedding watermarks derived from session keys into fingerprint images, linking user biometrics with device credentials. Tbatou et al. [23] tackled password-guessing vulnerabilities by integrating Diffie-Hellman key exchange with dynamic salt generation, providing stronger resistance against brute-force attacks. Kadhim et al. [24] introduced a scheme combining biometric data with dynamic virtual passwords, enhancing login security while minimizing public-key computation. In a step towards IoT-specific adaptation, Prapty et al. [17] proposed KESIC, which replaces traditional encryption with HMAC operations and introduces counter-based synchronization in place of timestamp dependency—optimizing Kerberos for resource-constrained IoT environments.

1.2 Problem Statement

With IoT's expansion, particularly in critical sectors like healthcare, ensuring secure yet lightweight authentication has become a significant challenge. Protocols such as MASK [11] have made notable strides using lightweight primitives like hash functions, PUFs, and XOR operations to enable secure communication in IoMT environments. While MASK defends against several common threats, including replay, MITM, and cloning attacks, it remains vulnerable to offline password-guessing and privileged insider threats—two highly critical attack vectors in medical systems [25, 26]. Furthermore, MASK suffers from session continuity issues due to identity update mechanisms, potentially causing legitimate authentication attempts to fail. These shortcomings highlight the need for more secure, context-aware authentication mechanisms for IoMT deployments.

In parallel, Kerberos remains a well-established protocol for authenticating users over traditional networks [12]. However, it was not designed for IoT ecosystems, where devices operate under severe resource constraints and lack reliable time synchronization. Kerberos's reliance on password-based authentication makes it susceptible to advanced attack techniques like Kerberoasting [18], Silver Ticket, and Golden Ticket attacks [19,20]. Additionally, its computationally intensive encryption and dependence on synchronized timestamps make it ill-suited for constrained IoT devices. These limitations call for a fundamental rethinking of Kerberos' architecture to adapt it for secure, scalable, and lightweight operation in modern distributed environments.

1.3 Proposed Solution

To address the security limitations of the MASK protocol [11] in IoMT environments, this report introduces *P-MASFEP* (security-enhanced PUF (Physical Unclonable Functions)-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI(Public Key Infrastructure)). The fuzzy extractor [27] strengthens resistance against offline password-guessing attacks by deriving stable cryptographic keys from noisy biometric inputs. Once the user's authenticity is verified, the protocol establishes a session key that is securely shared using PKI [28], thereby preventing privileged insider attacks. Combining biometric resilience with asymmetric cryptography, P-MASFEP ensures strong mutual authentication and secure communication tailored to resource-constrained and high-risk medical IoT environments.

To overcome the limitations of traditional Kerberos in IoT ecosystems, this report further proposes *KFLIT* (Kerberos with FIDO and Lightweight extension for the Internet of Things). As a foundational step, *KFI* (Kerberos with FIDO Integration) is developed by replacing password-based authentication with FIDO's passkey mechanism, thus eliminating vulnerabilities such as Kerberoasting [18], Silver Ticket [19], and Golden Ticket attacks [20]. *KFLIT* extends KFI by optimizing the protocol for resource-constrained IoT environments. It replaces computationally heavy encryption with lightweight HMAC and XOR operations to reduce processing overhead and employs counter-based synchronization to eliminate the need for real-time clocks. An attestation mechanism is introduced in the initial authentication phase to verify IoT device integrity before access is granted.

These enhancements make KFLIT a secure, scalable, and efficient authentication framework suited for diverse IoT applications.

1.4 Contributions

This report aims to design and evaluate lightweight and secure authentication protocols suitable for IoT environments. The contributions are divided into two parts based on the proposed protocols: *P-MASFEP* for IoMT security and *KFLIT* for general IoT authentication. The contributions of the report are as follows:

1. P-MASFEP

1. Investigate existing authentication protocols in the context of IoMT, focusing on PUF-based schemes, lightweight key exchanges, and privacy-preserving remote authentication.

2. Analyze the MASK protocol [11], highlighting its susceptibility to offline password guessing, privileged insider threats, and identity update failure.

3. Proposal of P-MASFEP, a lightweight authentication protocol that combines PUFs, fuzzy extractors, and PKI to establish secure session keys and defend against insider and password-based attacks [27, 28].

4. Validate P-MASFEP through informal security analysis, comparative performance evaluation with MASK, and formal protocol verification using the Scyther tool [29, 30].

2. KFLIT

1. Examine the limitations of traditional Kerberos authentication in IoT, particularly its reliance on passwords, timestamp synchronization, and encryption-heavy operations [17, 18, 31].

2. Proposal for KFI and KFLIT, Kerberos-based protocols that eliminate password dependence using FIDO and incorporate lightweight HMAC and XOR operations for secure authentication in IoT [32].

3. Evaluate KFLIT through informal security analysis, comparison with KESIC [17], and formal verification using the Scyther tool [29, 30].

1.5 Thesis Layout

This thesis is organized into six chapters. Chapter 2 presents the technical background. Chapter 3 introduces the proposed P-MASFEP protocol. Chapter 4 discusses the KFLIT protocols. Chapter 5 provides the results. Finally, Chapter 6 concludes the thesis with key insights and outlines potential directions for future work.

Chapter 2

Technical Background

This chapter presents the foundational technologies and security principles underpinning the design of the proposed authentication protocols, P-MASFEP and KFLIT. Given the unique challenges in securing resource-constrained IoT environments, it is essential to understand the tools and mechanisms that enable secure, lightweight, and scalable authentication.

2.1 Physically Unclonable Functions (PUFs)

Physically Unclonable Functions (PUFs) are entities embedded in physical structures (e.g., integrated chips) that use a Challenge-Response Pair (CRP) mechanism to generate a unique response based on inherent physical attributes of the silicon [33]. These unpredictable physical variations arise during manufacturing, making each PUF instance unique and resistant to duplication. PUFs exhibit properties similar to one-way hash functions—when the same challenge is fed as input to a PUF, it consistently generates the same response, ensuring repeatability under stable environmental conditions. Moreover, if the same challenge is applied to different devices, the responses will differ due to the distinct physical characteristics of each chip. This uniqueness provides a strong defense against cloning and physical attacks, making PUFs an excellent lightweight security primitive for IoT and embedded devices [34].

PUF-based authentication involves two main phases: enrollment and authentication. During the enrollment phase, the device's PUF interacts with the server, which issues a sequence of challenges and records the corresponding responses. These CRPs are securely stored in a database. Later, during authentication, the server sends a random challenge to the device. The device computes a response using its internal PUF and returns it to the server. If the response matches the enrolled value, the device is considered genuine. If not, the authentication fails. This CRP-based model enables lightweight, hardware-rooted authentication without storing sensitive keys on the device.

PUFs offer inherent resistance to physical and invasive attacks since the response behavior is deeply tied to uncontrollable silicon-level properties. Additionally, PUFs are highly advantageous in resource-constrained IoT environments due to their minimal hardware overhead and ability to generate volatile, on-demand secrets without relying on secure memory [35,36]. These features have made PUFs increasingly popular in secure embedded system design and lightweight cryptographic protocols.

2.2 Fuzzy Extractor (FE)

The fuzzy extractor (FE) is a cryptographic technique designed to resist offline password-guessing attacks by extracting robust cryptographic keys from noisy and non-deterministic biometric data [27]. Biometric traits such as fingerprints, iris scans, or voice patterns are inherently variable due to environmental factors, sensor inaccuracies, or user conditions. Unlike passwords or tokens, these biometrics cannot be reproduced identically each time. Therefore, a mechanism is needed to consistently extract the same secret from similar but not identical inputs—this is where fuzzy extractors prove essential. Without compromising security, FE can derive stable cryptographic keys from inputs close to, but not the same as, the original. It allows legitimate users to be authenticated even when the biometric reading slightly varies, thus supporting real-world usability while resisting brute-force attempts.

The FE consists of two core algorithms: Key Generation ($FE.Gen$) and Key Reconstruction ($FE.Rec$). In the key generation phase, the algorithm takes an original biometric input (BIO_i) and generates a secure cryptographic key (K) along with public helper data (h_d). This helper data is designed to leak minimal information about BIO_i, preserving the secrecy of the key. In the reconstruction phase, even if a slightly noisy version of the original biometric (BIO_i') is provided, the algorithm uses the stored h_d to regenerate the same key K ensuring robustness and resilience to noise in biometric readings, making FE ideal for scenarios like secure device authentication in IoMT and IoT systems.

Fuzzy extractors are increasingly being used in combination with Physically Unclonable Functions (PUFs) and public-key cryptography to build lightweight authentication mechanisms that are resistant to password-related threats and side-channel attacks, particularly in healthcare and embedded environments [34, 36].

2.3 Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI) ensures that data exchanged over untrusted networks, such as the Internet, remains secure and trustworthy. PKI provides a framework that enables encryption, digital signing, and identity verification through a combination of cryptographic elements and trust hierarchies. It uses components like digital certificates, certificate authorities (CAs), registration authorities (RAs), and key pairs to establish and verify the authenticity of communicating entities.

At the core of PKI lies the concept of asymmetric cryptography, which uses a public-private key pair. These keys are mathematically linked but computationally infeasible to derive one from the other [28,37]. The public key is openly shared and used for encryption or signature verification, while the private key is kept secret and used for decryption or signing. When users wish to send encrypted data, they retrieve the recipient's public key (typically embedded in a digital certificate issued by a trusted CA). This public key encrypts the data, resulting in a ciphertext that only the corresponding private key can decrypt. Since only the intended recipient possesses the private key, confidentiality is preserved. This same infrastructure also allows users to digitally sign messages, ensuring integrity and non-repudiation, which is critical in sensitive domains such as e-governance, healthcare, and secure device communication.

PKI is foundational in modern IoT and IoMT authentication schemes, especially those involving mutual authentication and secure session key establishment. By integrating PKI, systems can resist man-in-the-middle (MITM) attacks and privileged insider threats by confirming the legitimacy of every party involved in a transaction. Thus, PKI supports encryption and facilitates trusted identity binding in complex network environments, making it a cornerstone of secure communication architectures [28, 37].

2.4 Mutual Authentication and Key Establishment General Scheme

In a general IoMT-based services setting, a typical setup unfolds in two pivotal phases: user device and sensor node registration, mutual authentication, and session key establishment. Fig. 2.1 illustrates the general scheme.

In phase 1, the gateway initiates the registration process for both the user device and the sensor node. The user device retrieves its unique ID and prompts the user to enter a password. Upon receiving this information, the gateway registers the user's device and stores the password for future verification. The gateway sends a registration request to the sensor node, and the node responds with its unique ID. The gateway uses this ID to complete the node's registration.

In phase 2, the user is required to enter their password. The gateway verifies the user's identity by checking the accuracy of the password. After successful authentication, the user device sends its identity to the gateway. The gateway verifies the message to ensure the user's device is authentic. Similarly, the gateway sends its identity to the user device, which verifies the gateway's authenticity. Subsequently, the sensor node and server mutually authenticate each other. The gateway then generates a session key and safely distributes it to the user's device and the sensor node. This comprehensive structure ensures secure device registration, mutual authentication, and session key establishment, enabling safe communication.

2.5 Fast Identity Online (FIDO)

The Fast Identity Online (FIDO) authentication mechanism replaces traditional password-based authentication with a secure, public-key cryptography-based framework [32]. This passwordless approach enhances security and usability by mitigating risks such as phishing, credential theft, and replay attacks. The user device (UD) generates a unique public-private key pair during the registration phase. The private key ($FIDO\text{-}Priv_{UD}$) is securely stored in a hardware-based FIDO authenticator (e.g., USB token, biometric-enabled device), while the corresponding public key ($FIDO\text{-}Pub_{UD}$) is registered with the authenticator server (AS) and associated with the identity of UD.

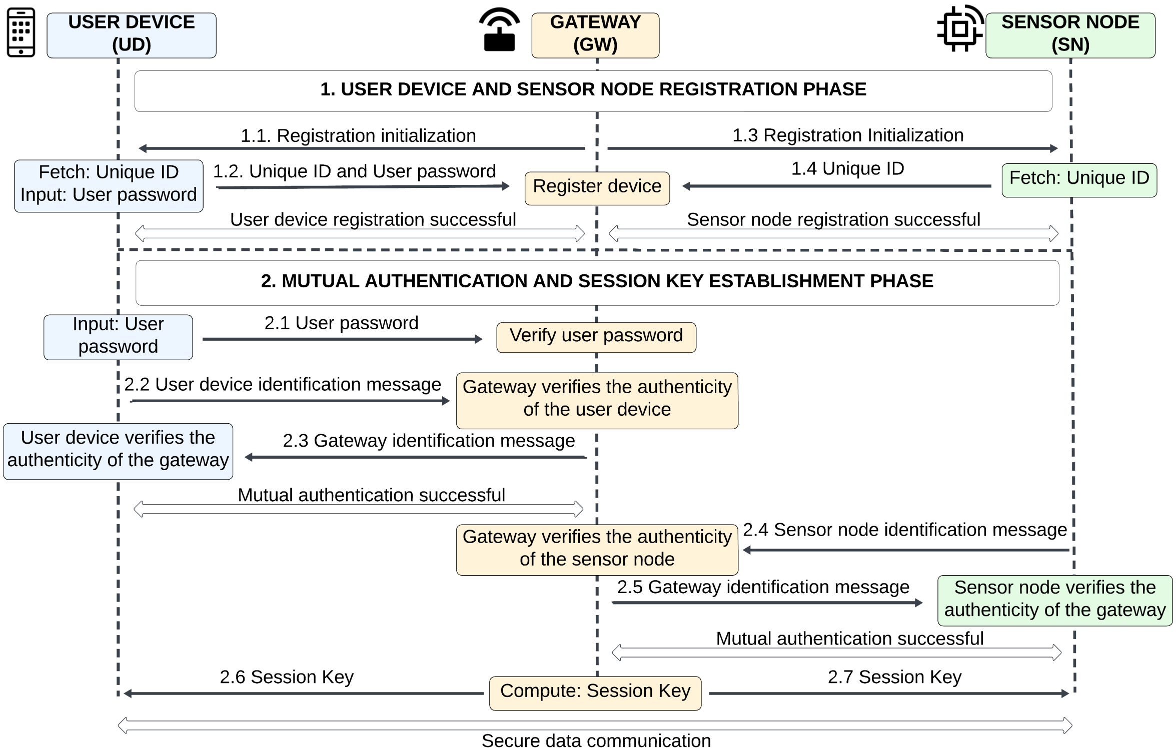


Figure 2.1: Mutual authentication and key establishment general scheme.

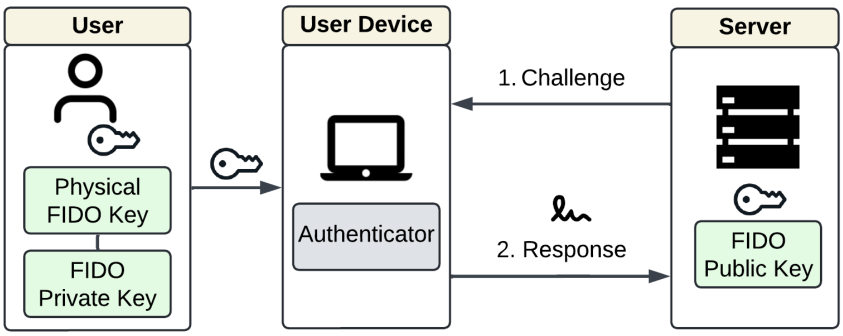


Figure 2.2: FIDO authentication process.

Fig. 2.2 illustrates the authentication process. AS issues a challenge to UD, which signs it using $FIDO\text{-}Priv_{UD}$ stored in the FIDO key. UD then sends the signed response back to AS, which verifies it using $FIDO\text{-}Pub_{UD}$. Successful verification authenticates UD securely and without transmitting any password.

1. ***Challenge:*** AS sends a cryptographic challenge ($Challenge_{AS,UD}$) to UD.

2. ***Response:*** The user inserts the physical FIDO key into UD, enabling the authenticator to compute a digital signature using $FIDO\text{-}Priv_{UD}$. UD sends the signed response back to AS, which verifies it using $FIDO\text{-}Pub_{UD}$. If the verification succeeds, AS authenticates UD.

This two-step authentication process—challenge issuance followed by signed response verification—ensures mutual trust without ever transmitting or storing user credentials. By replacing shared secrets with cryptographic key pairs, FIDO delivers a strong, phishing-resistant, scalable authentication framework. It is particularly well-suited for modern security architectures in distributed and resource-constrained environments such as IoT and enterprise systems [32].

2.6 Kerberos Authentication Protocol

Kerberos is a symmetric-key-based network authentication protocol designed to securely verify user identities over insecure networks [12]. It employs a trusted Key Distribution Center (KDC), composed of the Authenticator Server (AS) and Ticket Granting Server (TGS), to facilitate mutual authentication between a User Device (UD) and a Target Device (TD). The protocol ensures secure access by issuing time-limited encrypted tickets and never transmitting plaintext passwords across the network.

During the pre-registration phase, UD securely registers with KDC by sharing ID_{UD} and password. KDC derives a long-term symmetric key $K_{UD,AS}$ from the password, typically using a one-way cryptographic hash function. This key is securely stored and used later for authenticating UD without exposing the password over the network. However, this design is susceptible to offline password-guessing attacks. If an attacker captures an encrypted message protected by $K_{UD,AS}$, they can attempt to brute-force it offline. Fig. 2.3 illustrates the process. The notation used is summarised in Table 4.2. The Kerberos authentication process consists of the following steps:

1. ***Pre-Registration Phase:*** UD registers with KDC over a secure channel by submitting its identity (ID_{UD}) and password. KDC derives the long-term symmetric key ($K_{UD,AS}$ = Hash(Password)), which is securely stored and later used for authentication [38].

2. ***Authentication Request:*** To initiate authentication, UD sends an authentication request ($Req_{UD,AS} = ID_{UD} \parallel ID_{TGS} \parallel TS_{UD,AS}$) to AS. AS retrieves $K_{UD,AS}$ and generates a TGT and a session key ($k_{UD,TGS}$), then computes the ticket, $TGT = E_{K_{TGS,AS}}[k_{UD,TGS} \parallel ID_{UD} \parallel AD_{UD} \parallel ID_{TGS} \parallel L_2 \parallel TS_{AS,UD}]$ and $Res_{AS,UD} = E_{K_{UD,AS}}[TGT \parallel k_{UD,TGS}]$, which is sent to UD [39].

3. ***TGT Issuance:*** Upon receiving $Res_{AS,UD}$, UD decrypts it using $K_{UD,AS}$ to obtain $k_{UD,TGS}$ and TGT. TGT remains encrypted under $K_{TGS,AS}$, ensuring that UD cannot read or alter its contents [40].

4. ***Service Ticket Request:*** UD generates $A_{UD,TGS} = E_{k_{UD,TGS}}[ID_{UD} \parallel AD_{UD} \parallel TS_{UD,TGS}]$ and sends $Req_{UD,TGS} = ID_{TD} \parallel TGT \parallel A_{UD,TGS}$ to TGS, which decrypts and validates the TGT using $K_{TGS,AS}$ and the authenticator using $k_{UD,TGS}$ [40].

5. ***Service Ticket Issuance:*** TGS generates a session key ($k_{UD,TD}$) and constructs the service ticket ($T_{TD} = E_{K_{TD,TGS}}[k_{UD,TD} \parallel ID_{UD} \parallel AD_{UD} \parallel ID_{TD} \parallel TS_{TGS,UD} \parallel L_4]$) and response ($Res_{TGS,UD} = E_{k_{UD,TGS}}[T_{TD} \parallel k_{UD,TD}]$), which is sent to UD.

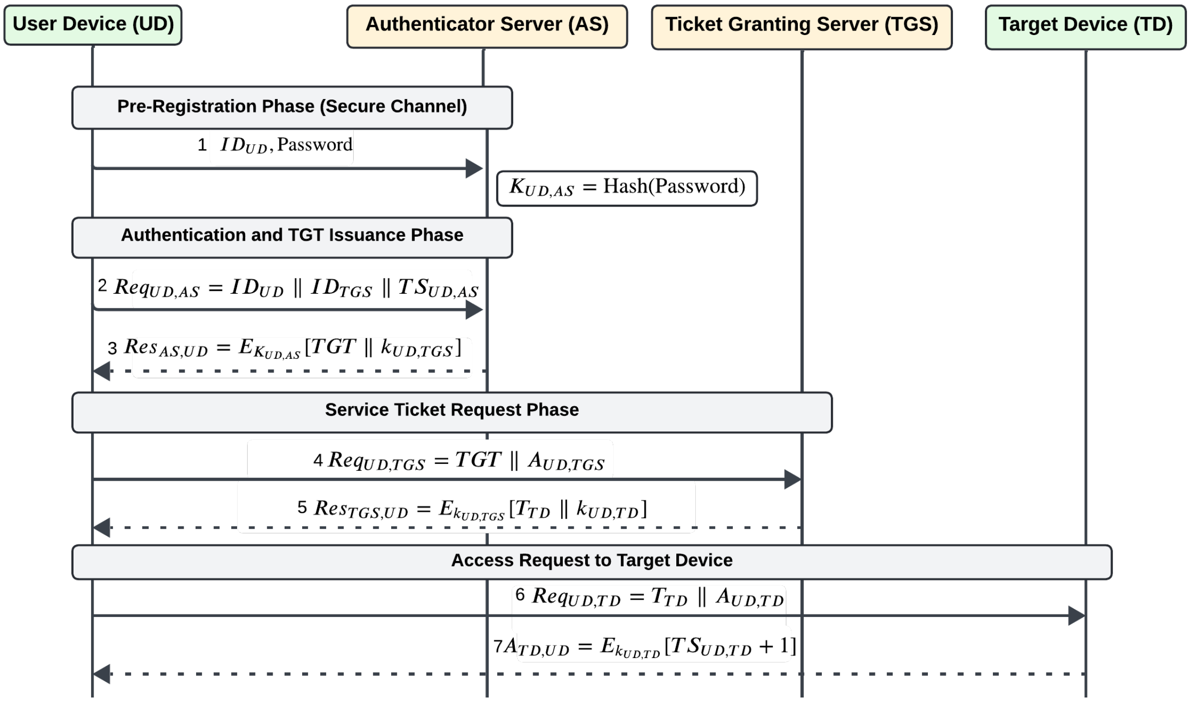


Figure 2.3: Kerberos steps.

6. ***Access Request to Target Device:*** UD decrypts $Res_{TGS,UD}$ to retrieve $k_{UD,TD}$ and T_{TD}, and constructs an authenticator ($A_{UD,TD} = E_{k_{UD,TD}}[TS_{UD,TD}]$). It then sends $Req_{UD,TD} = T_{TD} \parallel A_{UD,TD}$ to TD.

7. ***Target Device Validation and Access Granting:*** TD decrypts T_{TD} using $K_{TD,TGS}$ and verifies $A_{UD,TD}$ using $k_{UD,TD}$. Upon successful verification, it responds with $A_{TD,UD} = E_{k_{UD,TD}}[TS_{UD,TD} + 1]$ to confirm mutual authentication and grants UD access [40].

In summary, Kerberos provides a secure and efficient authentication protocol by leveraging symmetric key cryptography and time-bound tickets to enforce mutual trust and protect against credential theft. However, its dependence on password-derived keys exposes it to offline password-guessing attacks, particularly if long-term keys are compromised. Moreover, the reliance on synchronized timestamps and computationally intensive encryption operations can hinder its applicability in resource-constrained IoT environments. These limitations motivate the need for enhanced variants of Kerberos, such as the proposed KFI and KFLIT protocols, which strengthen authentication security while maintaining lightweight and scalable performance.

2.7 Scyther Protocol Verification Tool

Scyther is a protocol verification tool used to analyze and verify the security of cryptographic protocols. Cas Cremers [30] created it and used formal techniques, namely symbolic model checking, to evaluate a protocol's correctness and resilience against various security attributes. Using a high-level protocol description language, Scyther enables researchers and developers to define the protocol's behaviour, including its participants, messages sent, and security assumptions. Using automated analysis techniques, Scyther investigates every scenario to find potential security flaws, such as cryptographic flaws, protocol weaknesses, or vulnerabilities to particular attacks, such as impersonation, replay, or MITM attacks [29]. Scyther ensures that cryptographic protocols comply with security requirements and standards by thoroughly examining the protocol's behaviour and comparing it to predetermined security properties. This process yields essential insights into the reliability and efficacy of the protocols in actual deployment scenarios.

2.8 IoT vulnerabilities

1. ***Limited Processing Capabilities and Hardware Restrictions:*** IoT devices have limited processing capacity since manufacturers design them for specific purposes. There is little space left for adding strong security and data protection measures because of this restriction.

2. ***Heterogeneous Transmission Technology:*** Various communication methods are used by IoT devices to connect with the network. This variability makes creating uniform standards and safety precautions for every encounter difficult.

3. ***Gap in User Security Awareness:*** Many consumers are unaware of safeguarding their medical devices. Consumers can expose their devices to potential dangers without their knowledge.

4. ***Weak Physical Security:*** Many components of the IoT-based devices are physically accessible. This lack of physical security makes it more likely that items will be tampered with or that unauthorized individuals will gain access.

2.9 IoT Security Requirements

1. ***Confidentiality:*** One of the most important security goals is maintaining confidentiality since it creates guidelines and standards for limiting who can access what data. Sensitive medical data is better protected when only authorized parties can access important user data.

2. ***Integrity:*** Integrity is the preservation of the accuracy and dependability of data. Retaining data correctness and reliability requires maintaining integrity. Ensuring that instructions sent to the IoT devices or the data these devices receive remain authentic and unchanged.

3. ***Availability:*** It ensures that IoT functionalities are always available to authorized users and linked devices, regardless of when or where they choose to access them.

2.10 IoT Security Threats

1. ***Privileged insider attacks:*** In IoT systems, users with elevated access—such as administrators, service providers, or system integrators—may intentionally or

unintentionally misuse their privileges. Such actions can compromise system integrity, leak sensitive information, or turn off services. Mitigating these threats requires strict access control, role-based authentication, and continuous monitoring mechanisms.

2. ***Offline password guessing attacks:*** Offline attacks occur when an adversary captures encrypted authentication data and attempts to guess passwords without interacting with the target system in real-time. In IoT, where devices may store credentials locally, this can lead to unauthorized access, data leaks, or manipulation of device behaviour. Strong password policies, key derivation functions, and encryption help defend against such threats.

3. ***Replay attacks:*** A replay attack captures legitimate data transmissions and retransmits them to deceive the recipient into accepting duplicated messages. In IoT, this can result in repeated commands, unauthorized access, or data manipulation. Secure timestamping, nonces, and challenge-response mechanisms are essential to prevent these attacks.

4. ***Denial-of-Service (DoS) attacks:*** IoT devices are particularly vulnerable to DoS attacks, where high traffic or requests overwhelm device or network resources, causing service unavailability. Traffic rate limiting, anomaly detection, and robust network design are critical countermeasures [41].

5. ***Man-in-the-Middle (MITM) attacks:*** In MITM attacks, an adversary intercepts communication between two IoT entities, potentially modifying or injecting data to manipulate operations. Secure communication protocols, end-to-end encryption, and digital signatures are necessary defences.

6. ***Impersonation attacks:*** Attackers may spoof the identity of legitimate devices or users to gain unauthorized access to IoT systems. Mitigation strategies include strong mutual authentication, device attestation, and anomaly detection techniques.

7. ***Physical attacks:*** IoT devices are often deployed in uncontrolled or public environments, making them susceptible to tampering, side-channel attacks, or hardware-level exploitation. Adversaries may extract sensitive data or disrupt operations through direct physical access. Tamper-evident enclosures, secure boot mechanisms, and regular firmware updates are essential for protecting physical-layer security.

Chapter 3

Proposed P-MASFEP Protocol

This chapter presents P-MASFEP (PUF-based Mutual Authentication and Session key establishment using Fuzzy Extractor and PKI), a secure and lightweight authentication protocol for IoMT environments. The proposed protocol addresses critical vulnerabilities in existing schemes, including offline password guessing and privileged insider attacks. The chapter introduces the motivation behind developing P-MASFEP, outlines related work, and defines the system and adversary models. It then details the complete design of the protocol, including the registration phase and the mutual authentication with the session key establishment process.

3.1 Motivation

IoMT has recently seen tremendous growth and success. IoMT integrates the healthcare industry with the IoT ecosystem, which enables medical data creation, collection, transmission, and analysis by connecting different healthcare systems and sensors using various technologies such as Wi-Fi, Bluetooth, and cellular networks [1,2]. The patient's medical sensors collect and monitor physiological parameters, wirelessly transmitting the data to the physician's devices. Hence, the doctor can conduct a more thorough patient health assessment based on this data [42].

Although IoMT offers convenient healthcare services to patients, it is imperative to acknowledge that this technological advancement has introduced specific challenges, particularly in the security realm [3]. These encompass a spectrum of concerns, ranging from replay attacks, man-in-the-middle (MITM) attacks, impersonation, privileged insider threats, offline password guessing, and denial of service (DoS) to physical hijacking [4]. Among all possible threats, offline password guessing is the most common vulnerability in many IoMT-based networks for several reasons [25]. The availability of advanced tools, computational resources, and inadequate password regulations allows attackers to repeatedly try passwords without alerts or network shutdown [6]. Unauthorized access can result in several adverse events, including data theft, medical record tampering, and compromising patient privacy and safety [5]. Additionally, attackers may use password-guessed IoMT devices to launch DoS attacks, flooding networks or services with traffic and causing disruptions, resource exhaustion, or service interruptions [5].

Within the IoMT network, insiders like helper staff and administrators have authorized access to medical equipment. Since patients' health records are valuable and already have a foothold in the system, they are attractive candidates for conducting privileged insider attacks to steal patients' medical data [26]. This health data makes purchasing medications, obtaining medical attention, or submitting false medical claims possible. The integrity, confidentiality, and accessibility of patients' medical records and services are jeopardised by these security breaches [7]. Therefore, protecting the security and integrity of IoMT is paramount to prevent such malicious activities and ensure patient safety.

3.2 Objective

Masud et al. [11] suggested a lightweight Mutual Authentication and Secret Key establishment (MASK) protocol for protecting sensitive data in IoMT networks. MASK leverages lightweight cryptographic primitives to confirm the authenticity of the nodes before generating a session key. The scheme prevents many security threats like replay, MITM, cloning, and side-channel attacks.

However, MASK is vulnerable to offline password-guessing and privileged insider attacks, which can cause severe problems by granting unauthorized access to the patient's medical records and threatening life and privacy. The approach also has a device update difficulty. Due to this, even if the user is legitimate, the following protocol run will not be allowed to be executed.

To address the security vulnerabilities in the MASK [11] scheme, this thesis introduces P-MASFEP: security-enhanced PUF-based Mutual Authentication and Session key establishment using Fuzzy Extractor and PKI. The scheme leverages the fuzzy extractor [27], a cryptographic technique that generates secure cryptographic keys from biometrics to resist offline password-guessing attacks. Following lightweight mutual authentication, the session key is generated and securely transmitted to the authenticated physician using public-key cryptography [28], effectively preventing privileged insider attacks.

3.3 Related Work

Alladi et al. [9] proposed a A uthentication protocol using Resource-constrained Internet of Things devices (HARCI) to target healthcare networks that contain devices that are limited in their resources. The protocol uses PUFs to generate secure session keys for two-way authentication between patient devices, patient sensor nodes, and the healthcare cloud server. HARCI provides end-to-end authentication, using distinct session keys at each stage of the authentication process and using the various responses provided by PUFs as challenge inputs. Because the Internet of Things devices typically have limited memory capacity, battery life, and computational power, the protocol addresses the necessity for energy-efficient security solutions in these devices. The scheme is insecure against DoS attacks and faces scalability, resource constraints, security analysis, interoperability, and user privacy limitations.

Ashraf et al. [8] stated that IoT-based intelligent healthcare systems must incorporate secure remote user authentication, which makes it necessary to protect sensitive patient data and improve safe healthcare services in the Internet of Things era. Privacy and security must be appropriately maintained when adopting Internet of Things devices for remote patient monitoring. To lower the costs of calculation and transmission, the Lightweight Privacy-Preserving Remote User Authentication and Key Agreement Protocol solve these problems. For authorized users, such as clinical personnel and medical professionals, the protocol enhances security and allows them to access patient information securely. The protocol could benefit from extensive testing to assess its resilience against advanced cyber threats and attacks. The scheme cannot resist impersonation, privileged insider, cloning and side-channel attacks. Furthermore, exploring the integration of advanced encryption techniques could enhance the security and privacy aspects of the protocol, addressing potential vulnerabilities and ensuring robust protection of sensitive healthcare data.

Gaba et al. [10] introduced a Lightweight Key Exchange (LKE) based on certificate renewal for the rapidly increasing field of the Industrial Internet of Things (IIoT). Faster data accessibility, problem detection, performance analysis, and manager remote machine control are all made possible by Industry 4.0. Despite its advantages, it is risky since the Internet of Things nodes use unprotected wireless networks. The unprotected wireless channel provided many more opportunities for the illegal nodes to obtain information and take over the industrial machinery. Legitimate IoT nodes can exchange keys on a lightweight platform with LKE, which also forbids illegitimate usage. LKE uses lightweight Elliptic Curve Qu-Vanstone (ECQV) based implicit certificates to generate keys and establish confidence between entities. The scheme is not secure against cloning and side-channel attacks. Further investigation is required into the protocol's scalability for large-scale industrial networks, exploring potential vulnerabilities under different attack scenarios and evaluating the protocol's performance in real-world industrial environments.

Gope et al. [21] developed a physically safe, lightweight, anonymous mutual authentication system designed explicitly for Industrial Wireless Sensor Networks (IWSN). IWSNs are a new class of generic wireless sensor networks (WSNs) with limitations on coverage, energy consumption, security, and connectivity. However, security and privacy are two significant concerns because IWSN nodes are Internet-connected and situated in unattended environments with little human intervention. A user's ability to obtain real-time information from the chosen sensor nodes is essential for IWSN. This task requires a protocol for user authentication. The protocol utilizes PUFs and lightweight cryptographic primitives to provide private and secure user authentication in IWSNs. The protocol can increase the reliability and credibility of IWSNs by enhancing their efficiency, security, and privacy in real-time data access. The suggested protocol uses lightweight cryptographic primitives such as bitwise exclusive (XOR) operations, PUF, and one-way cryptographic hash functions. The scheme cannot withstand DoS and privileged insider attacks. According to a security and performance study, the proposed scheme is effective and safe for sensing devices in IWSN that have limited resources. The protocol's resilience to advanced attacks and scalability to accommodate growing sensor nodes must be investigated. Furthermore, the impact of hardware constraints and energy efficiency on the protocol's practical implementation in resource-constrained sensor nodes remains a significant research gap that requires attention.

Shao et al. [22] introduced a unique PUF-based anonymous authentication technique to protect patient data in wireless medical sensor networks (WMSNs). The protocol generates safe session keys while mutually authenticating doctors, sensors, and gateway nodes using cryptographic hash functions, fuzzy extractors, PUFs, and XOR operations. The technique regularly collects data from medical sensors and transports it via gateways to a monitoring centre, enabling real-time patient monitoring. The protocol may benefit from additional analysis and enhancements to address potential vulnerabilities related to insider attacks, desynchronization attacks, and sensor impersonation. Future research could focus on optimizing the communication and computation costs of the scheme to ensure efficient operation in resource-constrained WMSNs while maintaining a high level of security and privacy protection for sensitive medical data.

Masud et al. [11] proposed a lightweight and reliable Mutual Authentication and Secret Key (MASK) setup protocol for protecting sensitive health data in IoMT networks. Since the IoT nodes communicate critical data across the insecure wireless medium between virtual medical facilities, security is a significant concern in IoMT. This paper presents a lightweight, physically secure mutual authentication and secret key establishment protocol that employs PUF. PUF prevents side-channel, cloning, and manipulation attacks on sensor nodes deployed in unsupervised and hostile environments. Its design protects it from side-channel assaults, physical device loss, and security threats. It also ensures resource efficiency. The protocol uses lightweight encryption primitives such as bitwise XOR operations, nonces, PUFs, and one-way hash functions. The method thoroughly examines the adversary model, demonstrating the efficacy and efficiency of the MASK protocol in safeguarding IoMT networks. The scheme prevents many security threats, such as replay, MITM, cloning and side-channel attacks. However, MASK [11] is susceptible to offline password-guessing and privileged insider attacks, which can cause severe problems by granting unauthorized access to the patient's medical information and threatening life and privacy. The approach also has a device update difficulty. After each successful protocol run, the user's device will update the temporary identity. The device calculates the value used to verify the user's legitimacy using the expired identity. Therefore, even if the user is legitimate, the following protocol run will not be allowed to be executed.

Table 3.1: Comparison of authentication schemes for IoMT.

Scheme	**Objective**	**Key Features**
HARCI [9]	Secure healthcare authentication protocol for resource-constrained IoT devices	Three-layered architecture, utilizes PUFs for secure session key generation, separate session keys for each authentication phase
Ashraf et al. [8]	Ensure secure remote user authentication and key exchange in IoT-based healthcare systems	Symmetric session key exchange, lightweight solution, reduces computation and transmission costs, enhances security and privacy for remote patient monitoring
LKE [10]	Provides mutual authentication and secret key establishment for IIoT devices	Lightweight protocol, stresses certificate renewal, ensures message integrity, defends against attacks, preserves data privacy
Gope et al. [21]	Secures user authentication in IWSNs	Lightweight cryptographic primitives, utilizes PUFs for physical security, enhances security, privacy, and efficiency in IWSNs
Shao et al. [22]	Protects patient data in WSNs	Utilizes PUFs, cryptographic hash functions, fuzzy extractors, XOR operations, enables real-time patient monitoring
MASK [11]	Secures health information transfer in IoMT	Lightweight cryptography, one-way hash functions, nonces, PUFs, bit-wise XOR operations, prevents physical device loss, resilient against security threats, detailed system model and performance analysis

3.4 Limitations of the MASK Scheme

The MASK [11] scheme follows the general scheme discussed in Section 2.4. It prevents many security threats like replay, impersonation, cloning, and side-channel attacks. However, an adversary can effectively execute privileged insider and offline password-guessing attacks, which can expose sensitive information. Furthermore, MASK experiences a device update issue when logging in for the next session. The details are as follows.

3.4.1 Two Implementation Issues

In the MASK scheme, the user's device must provide the gateway with the sensor node's pseudo-identity to perform mutual authentication and secret session key establishment. However, during registration, the user's device does not record the sensor node's identity or take input from the user. As a result, it is impossible to complete the following steps. Furthermore, the user's device will update the temporary identity in the first protocol run. The user must input a password to prove its legitimacy in the second run. The user's device calculates the value used to verify the user's legitimacy using the expired identity. Therefore, the device will not permit the execution of the following protocol run, even if the user is authentic and enters the correct password. Thus, the protocol's implementation had flaws. Refer to the general scheme discussed in Section 2.4.

3.4.2 Offline Password Guessing Attack

In the MASK scheme, verifying a guessed password's accuracy is possible using data extracted from the user's device. Refer to Step 2.1 of the General Scheme discussed in Section 2.4. The attacker computes a value based on the guessed password and compares it with a value derived from the actual password and device data without triggering alerts or shutting down the network. If the values match, the attacker has successfully guessed the password, which makes the MASK scheme vulnerable to offline password-guessing attacks. In contrast, the proposed P-MASFEP scheme enhances security by using a fuzzy extractor to generate stable cryptographic keys from the user's biometrics, verifying the user's legitimacy. This process is detailed in Step 1 of the mutual authentication and session key establishment phase in Section 3.7.

3.4.3 Privileged Insider Attack

In the MASK scheme, a privileged insider with access to the gateway's resources could misuse their authority to compute the session key. During the mutual authentication and session key establishment phase, the insider could combine a unique user identifier with a constant extracted from the gateway's resources to calculate a specific value. The insider could eventually derive the session key using this value and another identifier linked to the user. In contrast, the proposed P-MASFEP scheme ensures that the sensor node generates the session key after mutual authentication. The sensor node then securely transmits the session key to the authenticated device using PKI, preventing privileged insiders from accessing the session key. This process is detailed in Steps 9 and 13 of the mutual authentication and session key establishment phase in Section 3.7.

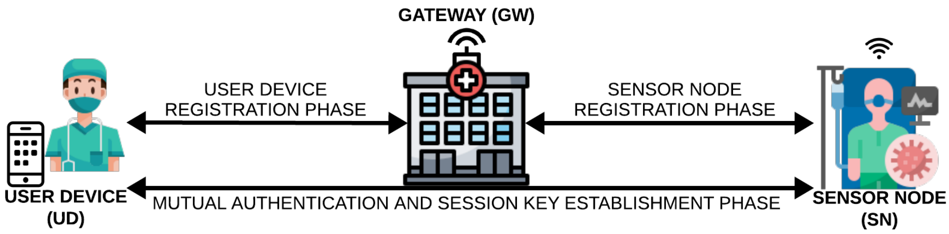


Figure 3.1: System model of P-MASFEP.

3.5 System Model

The User Device (UD), Gateway (GW), and Sensor Node (SN) comprise the system model depicted in Fig. 3.1.

1. ***User Device:*** The user connects to the sensor node to access real-time patient medical data, enabling quick patient care decisions.
2. ***Gateway:*** The gateway relays the patient's medical data, connecting the trusted user device and sensor node.
3. ***Sensor Node:*** The sensor nodes gather and transmit the patient's medical data to the user device via the gateway.

3.6 Adversary Model

The Dolev–Yao (DY) model [43] assumes an adversary with unlimited computational power and full access to all messages transmitted over the network. The adversary can eavesdrop, modify, delete, and insert messages on the public channel. In addition, the Canetti-Krawczyk (CK) adversary model [44] is employed to evaluate protocols with active adversaries who can manipulate messages and interact with honest participants. In this model, an attacker can also exploit power analysis attacks [45] to access the gateway's ephemeral parameters. The DY and CK adversary models account for many potential attacks. Further details are provided below.

- Ephemeral parameters of the gateway can be obtained by a privileged insider who can act as an adversary.
- The sensor node is vulnerable to physical assaults.
- An adversary can carry out several attacks, including impersonation, offline password guessing, privileged insider, replay, DoS, and MITM.

Table 3.2: Notations for P-MASFEP.

Notation	Definition
UD	User Device
GW	Gateway
SN	Sensor Node
P_{UD}, P_{SN}	P: Physically Unclonable Function, UD: User Device, SN: Sensor Node
C_{E}^{N}, R_{E}^{N}	C: Challenge, R: Response, E: Entity, N: Number
N_{UD}, N_{GW}, N_{SN}	N: Nonce, UD: User Device, GW: Gateway, SN: Sensor Node
TID_{UD}, TID_{SN}	TID: Temporary Identity, UD: User Device, SN: Sensor Node
D_{LN}	Unique License Number of the Doctor Issued by the Medical Council
D_{ID}	Unique Identity of the Doctor Issued by the Hospital
SN_{IEI}	International Equipment Identity of the Sensor Node
SK	Session Key
$A \equiv^{?} B$	Is A identical to B
$\oplus$, $\Vert$	Bit-wise XOR and Concatenation Operator
BIO_{i}	Biometrics of the i^{th} User
$FE.Gen(.)$	Fuzzy Extractor Generation Function
$FE.Rec(.)$	Fuzzy Extractor Reconstruction Function
K	Key
h_d	Helper Data
CPW_{i}	Computed Password of the i^{th} User
$PubK_{\text{UD}}$, $PubK_{\text{SN}}$	PubK: Public Key, UD: User Device, SN: Sensor Node
$PvtK_{\text{UD}}$, $PvtK_{\text{SN}}$	PvtK: Private Key, UD: User Device, SN: Sensor Node
$h(.)$	Cryptographic Hash Function
$E(.)$	Cryptographic Encryption Function
$D(.)$	Cryptographic Decryption Function

3.7 Proposed P-MASFEP Scheme Framework

The proposed P-MASFEP scheme extends the MASK [11] scheme, which follows the general scheme discussed in Section 2.4. P-MASFEP focuses on enhancing registration, mutual authentication, and session key establishment to address privileged insider and offline password-guessing attacks. Table 3.2 presents the notations utilized to explain the proposed scheme. The following assumptions are taken into account when creating the scheme:

1. UD and SN registration phase is completed using a secure channel.
2. UD, GW, and SN compute the same cryptographic functions to secure data by converting it into ciphertext. It makes it computationally challenging to reverse, ensuring the data's confidentiality, integrity, and authenticity.
3. UD, GW, and SN compute the same hash function. Properties include pre-image resistance (given a hash value, it should be computationally infeasible to find the original input), second pre-image resistance (given an input, it should be computationally infeasible to find another input with the same hash), and collision resistance (it should be computationally infeasible to find two different inputs with the same hash) [46].
4. GW is a reliable entity with sufficient processing power and storage.
5. UD and SN have enough processing power and storage to handle cryptographic functions.

3.7.1 User Device and Sensor Node Registration Phase

The following subsections present the stepwise explanation of the registration phase between the user device (UD), sensor node (SN), and gateway (GW).

User Device Registration Phase

Fig. 3.2 illustrates how the User Device (UD) registers with the Gateway (GW), establishing the initial trust and enabling subsequent authenticated interactions with the Sensor Node (SN). The registration leverages Physically Unclonable Functions (PUFs), user-specific identifiers, and biometric-enhanced credential generation for robust identity binding.

Step 1–2: The Gateway (GW) initiates the registration by generating a random PUF challenge and a set of synchronization challenges:

$$C_{UD}^{0}, C_{UD}^{SYN} \leftarrow \text{Random}() \tag{3.1}$$

GW constructs a registration message containing both challenges and sends it to UD:

$$Msg_{GW\to UD} = \{C_{UD}^{0}, C_{UD}^{SYN}\} \tag{3.2}$$

These challenges are stimuli to extract a device-unique response from UD's embedded PUF circuit.

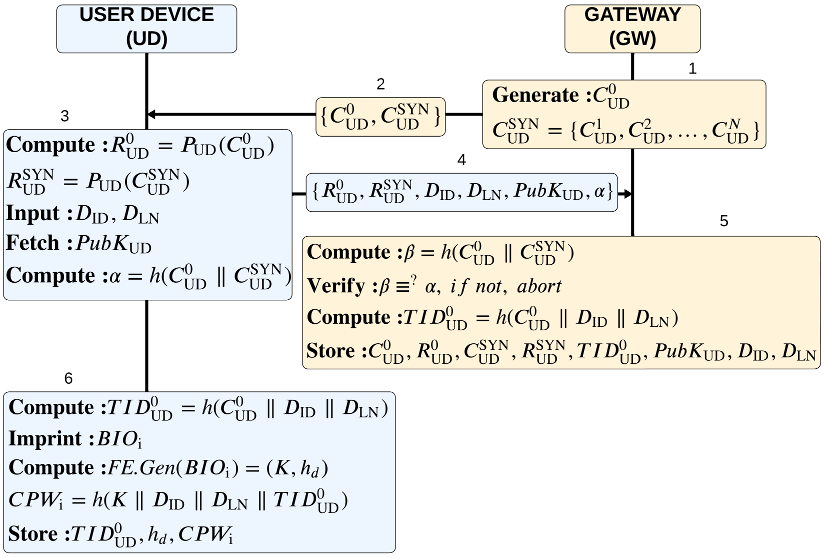


Figure 3.2: User device registration phase.

Step 3–4: Upon receiving the challenge set, UD computes two response tokens using its onboard PUF:

$$R^0_{UD} = PUF(C^0_{UD}) \tag{3.3}$$

$$R^{SYN}_{UD} = PUF(C^{SYN}_{UD}) \tag{3.4}$$

The user then inputs their unique identifiers: device ID (D_{ID}) and location/network ID (D_{LN}). UD also fetches its public key ($PubK_{UD}$) and computes a challenge digest:

$$\alpha = h(C^0_{UD} \parallel C^{SYN}_{UD}) \tag{3.5}$$

This digest enables GW to validate the integrity and pairing of the challenge-response pairs. UD then sends its registration request:

$$Msg_{UD \to GW} = \{R^0_{UD}, R^{SYN}_{UD}, D_{ID}, D_{LN}, PubK_{UD}, \alpha\} \tag{3.6}$$

Step 5: Upon receiving the message, GW performs the following operations:

→ It recomputes $\beta = h(C^0_{UD} \parallel C^{SYN}_{UD})$ and verifies whether $\beta \equiv \alpha$, confirming that the challenge and response pairings received from UD are valid.

→ If verification succeeds, GW binds the identity of UD by computing a unique temporary identifier:

$$TID^0_{UD} = h(C^0_{UD} \parallel D_{ID} \parallel D_{LN}) \tag{3.7}$$

→ Finally, GW securely stores: $\{C^0_{UD}, R^0_{UD}, C^{SYN}_{UD}, R^{SYN}_{UD}, TID^0_{UD}, PubK_{UD}, D_{ID}, D_{LN}\}$

Step 6: In parallel, UD computes the same temporary identity locally:

$$TID^0_{UD} = h(C^0_{UD} \parallel D_{ID} \parallel D_{LN}) \tag{3.8}$$

UD then performs biometric enrollment. It captures the biometric imprint BIO_i and applies a fuzzy extractor to derive:

$$(K, h_d) = FE.Gen(BIO_i) \tag{3.9}$$

This pair (K, h_d) enables reproducible key generation in future authentications. Next, UD derives the credential:

$$CPW_i = h(K \parallel D_{ID} \parallel D_{LN} \parallel TID^0_{UD}) \tag{3.10}$$

It securely stores TID^0_{UD}, h_d, and CPW_i for future use during mutual authentication.

This registration phase establishes a strong device identity at GW and securely binds it to user-specific and hardware-specific secrets on UD. The combination of PUF, biometrics, and cryptographic hashing enhances resistance against cloning, impersonation, and offline attacks.

Sensor Node Registration Phase

This phase enables the Sensor Node (SN) to securely register with the Gateway (GW) using PUF-based authentication and unique node identity parameters. It establishes a trusted identity for SN in the system, allowing it to participate in future authenticated communication with user devices (UD). Fig. 3.3 depicts the detailed registration steps.

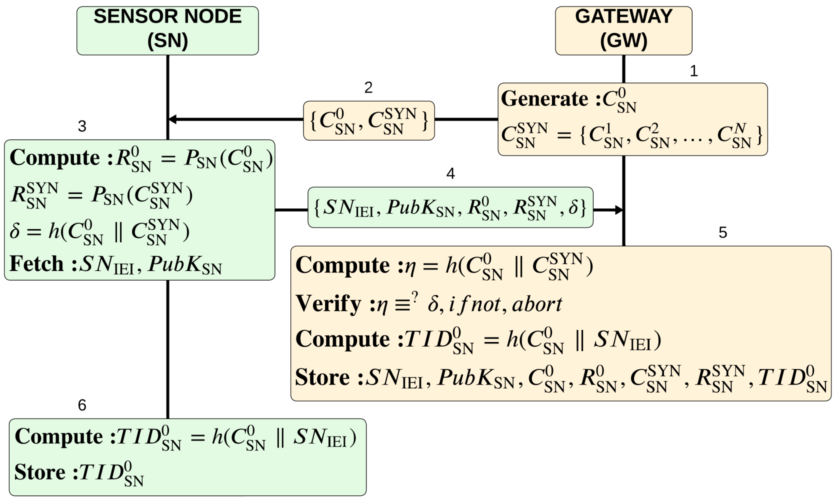


Figure 3.3: Sensor node registration phase.

Step 1–2: The Gateway (GW) begins SN registration by generating a unique challenge for the PUF along with a synchronization challenge set:

$$C^0_{SN},\ C^{SYN}_{SN} \leftarrow \text{Random}() \tag{3.11}$$

It transmits the challenge bundle to SN:

$$Msg_{GW \to SN} = \{C^0_{SN}, C^{SYN}_{SN}\} \tag{3.12}$$

These challenges are designed to extract device-specific fingerprints from the SN using its physical PUF circuit.

Step 3–4: Upon receiving the challenges, SN computes the corresponding responses:

$$R^0_{SN} = PUF(C^0_{SN}) \tag{3.13}$$
$$R^{SYN}_{SN} = PUF(C^{SYN}_{SN}) \tag{3.14}$$
$$\delta = h(C^0_{SN} \parallel C^{SYN}_{SN}) \tag{3.15}$$

It then prepares a registration message containing:

→ SN_{IEI}: The Sensor Node's Identity and Environment Information (e.g., hardware or deployment profile).

→ $PubK_{SN}$: SN's public key for future encrypted communication.

→ The PUF responses and challenge digest δ for integrity verification.

SN sends this message to GW:

$$Msg_{SN \to GW} = \{SN_{IEI}, PubK_{SN}, R_{SN}^0, R_{SN}^{SYN}, \delta\} \tag{3.16}$$

Step 5: GW verifies SN's response by recalculating the challenge digest:

$$\eta = h(C_{SN}^0 \parallel C_{SN}^{SYN}) \tag{3.17}$$

It compares η with the received δ. If they match, GW proceeds to bind SN's identity to the challenge by computing:

$$TID_{SN}^0 = h(C_{SN}^0 \parallel SN_{IEI}) \tag{3.18}$$

This temporary identifier ensures unique and traceable registration of the SN in the system. GW stores the complete record: $\{SN_{IEI}, PubK_{SN}, C_{SN}^0, R_{SN}^0, C_{SN}^{SYN}, R_{SN}^{SYN}, TID_{SN}^0\}$ for future reference.

Step 6: SN independently computes the same temporary identifier for local storage and future use:

$$TID_{SN}^0 = h(C_{SN}^0 \parallel SN_{IEI}) \tag{3.19}$$

This identifier binds the SN's physical identity (via PUF) and its environmental tag (SN_{IEI}), ensuring that only the legitimate node can later authenticate itself.

This registration process securely links the physical and digital identity of the sensor node, making it resilient to cloning and impersonation attacks. By storing both the PUF responses and key metadata, GW can verify the authenticity of SN later during mutual authentication.

3.7.2 Mutual Authentication and Session Key Establishment Phase

This phase ensures secure mutual authentication between the user device (UD) and sensor node (SN) through the gateway (GW), and establishes a session key for encrypted communication. The process combines biometric-based credential verification, PUF-based device authentication, and public key encryption for confidentiality. Fig. 3.4 presents the overall message flow.

Step 1–2: The user must first prove the identity by entering D_{ID}, D_{LN} and imprint BIO_i. P-MASFEP makes it harder for an attacker to guess the password by using biometric information BIO_i to calculate:

$$K^* = \text{FE.Rec}(BIO_i, h_d) \tag{3.20}$$

$$CPW_i^* = h(K^* \parallel D_{ID} \parallel D_{LN} \parallel TID_{UD}^0) \tag{3.21}$$

If $CPW_i^* \equiv CPW_i$, UD generates a nonce N_{UD}^1 and computes:

$$N_{UD}^{1*} = N_{UD}^1 \oplus D_{ID} \tag{3.22}$$

$$TID_{UD}^{0*} = TID_{UD}^0 \oplus D_{LN} \tag{3.23}$$

UD composes a message containing $\{N_{UD}^{1*}, TID_{UD}^{0*}\}$ and transmits it to GW.

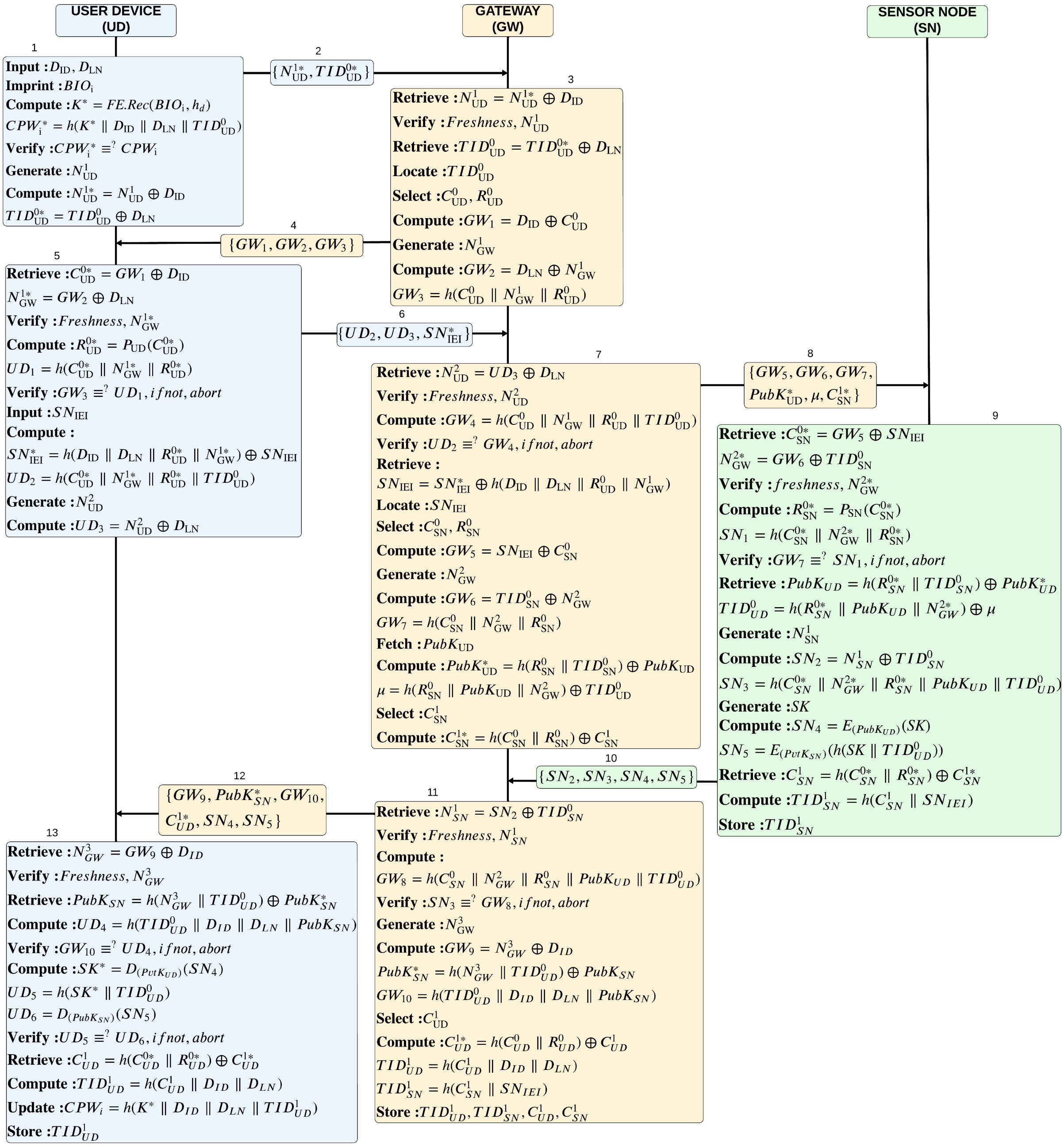


Figure 3.4: Mutual authentication and session key establishment phase.

Step 3–4: GW extracts the real nonce, $N_{UD}^1 = N_{UD}^{1*} \oplus D_{ID}$. Next, GW checks whether the nonce is fresh; if not, it aborts. GW derives the temporary identity, $TID_{UD}^0 = TID_{UD}^{0*} \oplus D_{LN}$ and matches with the database. If found, the authenticity of UD is proven; if not, it is a bogus request. Then GW selects the corresponding CRP, (C_{UD}^0, R_{UD}^0) and encloses the real C_{UD}^0 in:

$$GW_1 = D_{ID} \oplus C_{UD}^0 \tag{3.24}$$

$$GW_2 = D_{LN} \oplus N_{GW}^1 \tag{3.25}$$

$$GW_3 = h(C_{UD}^0 \parallel N_{GW}^1 \parallel R_{UD}^0) \tag{3.26}$$

GW sends $\{GW_1, GW_2, GW_3\}$ to UD.

Step 5–6: UD computes:

$$C_{UD}^{0*} = GW_1 \oplus D_{ID} \tag{3.27}$$

$$N_{GW}^{1*} = GW_2 \oplus D_{LN} \tag{3.28}$$

$$R_{UD}^{0*} = P_{UD}(C_{UD}^{0*}) \tag{3.29}$$

$$UD_1 = h(C_{UD}^{0*} \parallel N_{GW}^{1*} \parallel R_{UD}^{0*}) \tag{3.30}$$

If $UD_1 \equiv GW_3$, GW is authenticated. UD creates a pseudo-identity for SN and computes:

$$SN_{IEI}^* = h(D_{ID} \parallel D_{LN} \parallel R_{UD}^{0*} \parallel N_{GW}^{1*}) \oplus SN_{IEI} \tag{3.31}$$

$$UD_2 = h(C_{UD}^{0*} \parallel N_{GW}^{1*} \parallel R_{UD}^{0*} \parallel TID_{UD}^0) \tag{3.32}$$

$$UD_3 = N_{UD}^2 \oplus D_{LN} \tag{3.33}$$

UD sends $\{UD_2, UD_3, SN_{IEI}^*\}$ to GW.

Step 7–8: GW retrieves $N_{UD}^2 = UD_3 \oplus D_{LN}$ and computes:

$$GW_4 = h(C_{UD}^0 \parallel N_{GW}^1 \parallel R_{UD}^0 \parallel TID_{UD}^0) \tag{3.34}$$

$$SN_{IEI} = SN_{IEI}^* \oplus h(D_{ID} \parallel D_{LN} \parallel R_{UD}^0 \parallel N_{GW}^1) \tag{3.35}$$

Verifies $UD_2 \equiv GW_4$. GW selects CRP (C_{SN}^0, R_{SN}^0) and nonce N_{GW}^2 and computes:

$$GW_5 = SN_{IEI} \oplus C_{SN}^0 \tag{3.36}$$

$$GW_6 = TID_{SN}^0 \oplus N_{GW}^2 \tag{3.37}$$

$$GW_7 = h(C_{SN}^0 \parallel N_{GW}^2 \parallel R_{SN}^0) \tag{3.38}$$

$$PubK_{UD}^* = h(R_{SN}^0 \parallel TID_{SN}^0) \oplus PubK_{UD} \tag{3.39}$$

$$\mu = h(R_{SN}^0 \parallel PubK_{UD} \parallel N_{GW}^2) \oplus TID_{UD}^0 \tag{3.40}$$

$$C_{SN}^{1*} = h(C_{SN}^0 \parallel R_{SN}^0) \oplus C_{SN}^1 \tag{3.41}$$

GW sends $\{GW_5, GW_6, GW_7, PubK_{UD}^*, \mu, C_{SN}^{1*}\}$ to SN.

Step 9–10: SN computes:

$$C_{SN}^{0*} = GW_5 \oplus SN_{IEI} \tag{3.42}$$

$$N_{GW}^{2*} = GW_6 \oplus TID_{SN}^0 \tag{3.43}$$

$$R_{SN}^{0*} = P_{SN}(C_{SN}^{0*}) \tag{3.44}$$

$$SN_1 = h(C_{SN}^{0*} \parallel N_{GW}^{2*} \parallel R_{SN}^{0*}) \tag{3.45}$$

Verifies $SN_1 \equiv GW_7$. Then retrieves and computes:

$$PubK_{UD} = h(R_{SN}^{0*} \parallel TID_{SN}^0) \oplus PubK_{UD}^* \quad (3.46)$$
$$TID_{UD}^0 = h(R_{SN}^{0*} \parallel PubK_{UD} \parallel N_{GW}^{2*}) \oplus \mu \quad (3.47)$$
$$SN_2 = N_{SN}^1 \oplus TID_{SN}^0 \quad (3.48)$$
$$SN_3 = h(C_{SN}^{0*} \parallel N_{GW}^{2*} \parallel R_{SN}^{0*} \parallel PubK_{UD} \parallel TID_{UD}^0) \quad (3.49)$$
$$SN_4 = E_{PubK_{UD}}(SK) \quad (3.50)$$
$$SN_5 = E_{PvtK_{SN}}(h(SK \parallel TID_{UD}^0)) \quad (3.51)$$
$$C_{SN}^1 = h(C_{SN}^{0*} \parallel R_{SN}^{0*}) \oplus C_{SN}^{1*} \quad (3.52)$$
$$TID_{SN}^1 = h(C_{SN}^1 \parallel SN_{IEI}) \quad (3.53)$$

SN sends $\{SN_2, SN_3, SN_4, SN_5\}$ to GW.

Step 11–12: GW authenticates SN:

$$N_{SN}^1 = SN_2 \oplus TID_{SN}^0 \quad (3.54)$$
$$GW_8 = h(C_{SN}^0 \parallel N_{GW}^2 \parallel R_{SN}^0 \parallel PubK_{UD} \parallel TID_{UD}^0) \quad (3.55)$$

Verifies $SN_3 \equiv GW_8$. Then generates nonce N_{GW}^3 and computes:

$$GW_9 = N_{GW}^3 \oplus D_{ID} \quad (3.56)$$
$$PubK_{SN}^* = h(N_{GW}^3 \parallel TID_{UD}^0) \oplus PubK_{SN} \quad (3.57)$$
$$GW_{10} = h(TID_{UD}^0 \parallel D_{ID} \parallel D_{LN} \parallel PubK_{SN}) \quad (3.58)$$
$$C_{UD}^{1*} = h(C_{UD}^0 \parallel R_{UD}^0) \oplus C_{UD}^1 \quad (3.59)$$
$$TID_{UD}^1 = h(C_{UD}^1 \parallel D_{ID} \parallel D_{LN}) \quad (3.60)$$

GW sends $\{GW_9, PubK_{SN}^*, GW_{10}, C_{UD}^{1*}, SN_4, SN_5\}$ to UD.

Step 13: UD reconstructs session credentials:

$$N_{GW}^3 = GW_9 \oplus D_{ID} \quad (3.61)$$
$$PubK_{SN} = h(N_{GW}^3 \parallel TID_{UD}^0) \oplus PubK_{SN}^* \quad (3.62)$$
$$UD_4 = h(TID_{UD}^0 \parallel D_{ID} \parallel D_{LN} \parallel PubK_{SN}) \quad (3.63)$$
$$SK^* = D_{PvtK_{UD}}(SN_4) \quad (3.64)$$
$$UD_5 = h(SK^* \parallel TID_{UD}^0) \quad (3.65)$$
$$UD_6 = D_{PubK_{SN}}(SN_5) \quad (3.66)$$

If $UD_5 \equiv UD_6$, authentication is complete. UD then updates:

$$C_{UD}^1 = h(C_{UD}^{0*} \parallel R_{UD}^{0*}) \oplus C_{UD}^{1*} \quad (3.67)$$
$$TID_{UD}^1 = h(C_{UD}^1 \parallel D_{ID} \parallel D_{LN}) \quad (3.68)$$
$$CPW_i = h(K^* \parallel D_{ID} \parallel D_{LN} \parallel TID_{UD}^1) \quad (3.69)$$

Chapter 4

Proposed KFLIT Protocol

This chapter presents *KFLIT* protocol, an enhanced authentication framework designed to address the limitations of traditional Kerberos in password security and IoT applicability. *KFLIT* builds upon a foundational protocol, *KFI*, by eliminating password dependencies through FIDO's passwordless mechanism. It further extends *KFI* to suit resource-constrained IoT environments by incorporating lightweight cryptographic operations, counter-based synchronization, and device attestation. This chapter introduces the motivation for developing *KFLIT*, discusses related Kerberos-based schemes, identifies prevailing research gaps, and outlines the system and adversary models. The final sections detail the *KFI* and *KFLIT* protocols step by step.

4.1 Motivation

Kerberos is a widely adopted authentication protocol originally designed for traditional enterprise networks [12,14]. It provides mutual authentication using a ticket-based framework grounded in symmetric key cryptography. Despite its success in centralized and well-resourced environments, Kerberos faces growing limitations in modern security contexts.

A primary concern is Kerberos' reliance on user-chosen passwords, which exposes it to password-centric threats such as offline password guessing, Kerberoasting, Silver Ticket, and Golden Ticket attacks [18–20]. These attacks have been exploited in real-world intrusions, including APT20 [47] and FIN7 [48], highlighting critical weaknesses in Kerberos' password-dependent architecture.

Furthermore, Kerberos is not well-suited for IoT environments. Devices in IoT networks typically have constrained resources — limited processing power, memory, and energy — and often operate without stable real-time clocks. Kerberos' dependence on synchronized timestamps, frequent ticket exchanges, and encryption-heavy operations creates significant overhead that exceeds the capabilities of most IoT nodes [17, 49]. Although efforts like KESIC [17] have attempted to adapt Kerberos by introducing lightweight primitives such as HMAC and counter-based synchronization, they retain several inherent limitations. These include continued reliance on password-based logins, lack of built-in device integrity verification, and challenges in scaling across diverse and heterogeneous IoT environments.

These limitations underline the urgent need for an authentication protocol that retains Kerberos' strengths in mutual authentication while overcoming its unsuitability for passwordless and resource-constrained systems.

4.2 Objective

Although Kerberos avoids password transmission, it still depends on the secrecy and strength of user-chosen passwords. Weak passwords can be exploited through offline dictionary or brute-force attacks, leading to several advanced threats such as Kerberoasting [18], Golden Ticket attacks [20], and Silver Ticket attacks [19]. Furthermore, adapting Kerberos for IoT environments introduces new challenges due to device constraints and protocol design. Based on these observations, the following research gaps are identified:

1. ***Vulnerability to password-based attacks:*** Using password-derived keys in Kerberos makes it susceptible to brute-force, dictionary, and ticket forgery attacks.

2. ***Limitations in IoT environments:*** Traditional Kerberos requires clock synchronization and performs heavyweight cryptographic operations, making it inefficient for resource-constrained IoT devices.

To overcome these challenges, this thesis proposes two enhancements. The first, *KFI* (Kerberos with FIDO Integration), eliminates password-based login by incorporating FIDO's asymmetric passkey authentication, thereby addressing Kerberos' vulnerability to password-derived attacks. The second, *KFLIT* (Kerberos with FIDO and Lightweight Extension for IoT), extends KFI to make it compatible with IoT environments. KFLIT replaces encryption-heavy operations with lightweight primitives such as HMAC and XOR, employs counter-based synchronization to remove timestamp dependency, and integrates device attestation to verify the integrity of participating IoT devices before granting access. Together, these enhancements provide a secure, scalable, and resource-efficient authentication solution tailored for the evolving landscape of IoT.

4.3 Related Work

Several works in the literature have explored adaptations of the Kerberos protocol to enhance authentication in diverse environments. These include schemes integrating public-key, biometric, or blockchain-based mechanisms. Table 4.1 summarizes existing Kerberos-based approaches' key authentication mechanisms, features, and limitations.

Neuman et al. [12] introduced Kerberos as a secure authentication service designed for open network systems. It relies on symmetric cryptography and a trusted third party to establish secure authentication between clients and servers. The protocol mitigates the exposure of sensitive information by using encrypted tickets. However, Kerberos remains vulnerable to password-based attacks because it relies on user-chosen passwords. Additionally, the requirement for synchronized clocks and the computational overhead of encryption operations present challenges in resource-constrained environments.

Downnard [16] introduced public-key cryptography enhancements to the Kerberos protocol, addressing its reliance on password-based authentication and enhancing security and scalability. One of the key modifications is PKINIT (Public-Key-Based Initial Authentication), which replaces password-based authentication with public-key cryptography in the initial login. In PKINIT, the Kerberos server authenticates users through digital signatures and encrypts the TGT and session key with the user device's public key, safeguarding credentials against interception. Additionally, PKCROSS (Public-Key-Based Cross-Realm Authentication) enables secure cross-realm authentication by using public-key cryptography for KDC-to-KDC communication, removing the need for pre-shared symmetric keys across realms and thus streamlining multi-realm configurations. Another modification, PKDA (Public-Key-Based Distributed Authentication), extends the protocol by enabling direct client-to-server authentication using public-key signatures, which reduces the load on the centralized KDC, minimizes network bottlenecks, and enhances privacy. These enhancements mitigate Kerberos' susceptibility to password-related attacks, strengthen inter-realm security, and enable more flexible, distributed authentication across networks.

Tbatou et al. [23] proposed a mutual Kerberos authentication protocol for distributed systems, enhancing the security of the traditional Kerberos V5 by integrating the Diffie-Hellman key exchange and a dynamic salt generator for robust key management. The protocol operates in three phases: registration, communication, and renewal. In the registration phase, the client and KDC use Diffie-Hellman to securely generate initial authentication parameters with dynamic salts, ensuring unique session keys per User. The communication phase further enhances security with functions named S2KexS and DKexS, which use password-derived keys and dynamic salts for per-session encryption, making it more resilient against dictionary and brute-force attacks. Finally, the renewal phase allows for the periodic updating of client authentication parameters, enhancing security by preventing long-term exposure of sensitive data. This approach improves Kerberos' robustness against password-related attacks while ensuring a secure channel for distributed systems.

Han et al. [15] proposed a biometric-Kerberos authentication protocol explicitly designed for secure mobile computing services. The protocol combines traditional Kerberos authentication with biometric data to enhance security, especially for mobile devices in m-commerce applications. The scheme combines fingerprint biometrics captured through a smartphone camera with a watermark generated from the device's unique serial number. The watermark is embedded into the fingerprint image at the acquisition time, linking the biometric data to the specific device. The watermark embedding key is derived from the Kerberos session key, and only the trusted KDC can remove the watermark. This approach enables forensic traceability, ensuring that only a legitimate device and user can authenticate, as the watermark needs to be removed for a successful biometric match. By integrating this watermark with the biometric data, the protocol provides a cost-effective, high-security solution suitable for mobile environments with higher security risks due to device portability and the potential for interception on wireless networks.

Chen et al. [50] proposed DKSM, a decentralized protocol extending Kerberos for secure service management in IoT environments. DKSM innovatively combines blockchain technology and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to address Kerberos' limitations, such as single-point failure and replay attacks. The protocol decentralizes key distribution using smart contracts for transparent and immutable interactions. DKSM enhances privacy with fine-grained access control and robust nonce-based time synchronization. Experimental results on Ethereum and FISCO platforms validate their cost-efficiency and scalability. By integrating blockchain's traceability with Kerberos' authentication framework, DKSM provides a resilient and adaptable solution for distributed IoT ecosystems.

Gaikwad et al. [49] introduced a smart home automation system leveraging IoT and a robust three-level Kerberos authentication mechanism to enhance security. The proposed system integrates GSM/GPRS modules, RF communication, and microcontroller-based hardware for reliable and low-cost automation. Kerberos authentication is employed to mitigate security vulnerabilities, with three layers ensuring password protection, ticket-based authentication, and session management. The study demonstrates a seamless and secure process for monitoring and controlling household appliances, highlighting the system's scalability and resilience. The proposed solution addresses IoT-specific challenges like real-time communication and device integration while ensuring confidentiality and integrity in distributed environments.

Prapty et al. proposed *KESIC* [17], a lightweight Kerberos-based protocol for IoT environments. While *KESIC* introduces HMAC-based ticket generation and counter-based synchronization to reduce computational and memory overhead, it still relies on symmetric cryptography for authentication and key exchange. This dependency increases computational complexity and resource consumption, making it less suitable for highly constrained IoT devices. Moreover, *KESIC* retains Kerberos' password-based authentication, leaving it vulnerable to brute-force and dictionary attacks.

Table 4.1: Comparison of Kerberos-based authentication schemes.

Scheme	Authentication Mechanism	Key Features	Limitations
Neuman et al. [12]	Password-based authentication	Symmetric cryptography; Ticket-based authentication; Avoids password transmission over network	Vulnerable to brute-force and dictionary attacks; Requires clock synchronization; Computationally heavy
Downnard [16]	Public-key authentication	Eliminates password dependency; Supports cross-realm authentication	High computational overhead; Unsuitable for IoT
Tbatou et al. [23]	Password-based with Diffie-Hellman key exchange	Secure session keys; Periodic authentication updates	Increased complexity; Frequent key updates reduce efficiency
Han et al. [15]	Biometric authentication	Fingerprint-based authentication with watermarking	Requires biometric hardware; Vulnerable to biometric spoofing
Chen et al. [50]	Blockchain-based authentication	Decentralized key management with CP-ABE encryption	High reliance on blockchain; Latency in large-scale networks
Gaikwad et al. [49]	Ticket-based authentication	Three-layer authentication; Real-time monitoring	Struggles with scalability in dynamic IoT environments
KESIC [17]	Password-based authentication	Lightweight Kerberos adaptation; Counter-based synchronization	Vulnerable to password attacks; Still relies on symmetric cryptography, increasing computational overhead

Table 4.2: Notations for KFLIT.

Notation	Definition
	Entities and Components
UD	User Device
KDC	Key Distribution Center
AS	Authenticator Server
TGS	Ticket Granting Server
TD	Target Device
ITS	IoT Server
IN	IoT Node
	Keys and Cryptographic Operations
$K_{x,y}$	Long-term shared key between x and y
$k_{x,y}$	Short-term (session) key between x and y
$FIDO\text{-}Pub_{UD}$	User Device's FIDO Public Key
$FIDO\text{-}Priv_{UD}$	User Device's FIDO Private Key
$E_K[\cdot]$	Symmetric Encryption using key K
$D_K[\cdot]$	Symmetric Decryption using key K
$HMAC(K, M)$	HMAC operation with key K and message M
	Authentication and Ticketing
ID_x	Identity of Entity x
AD_{UD}	Network Address of the User Device
$TS_{x,y}$	Timestamp sent from x to y
L_n	Ticket Expiration Timestamp
T_x	Ticket for Entity x
TGT	Ticket Granting Ticket
$Req_{x,y}$	Request sent from x to y
$Res_{x,y}$	Response sent from x to y
$A_{x,y}$	Authenticator sent from x to y
$Challenge$	Challenge
$Attest$	Attestation phase operation
$Serv$	Service request

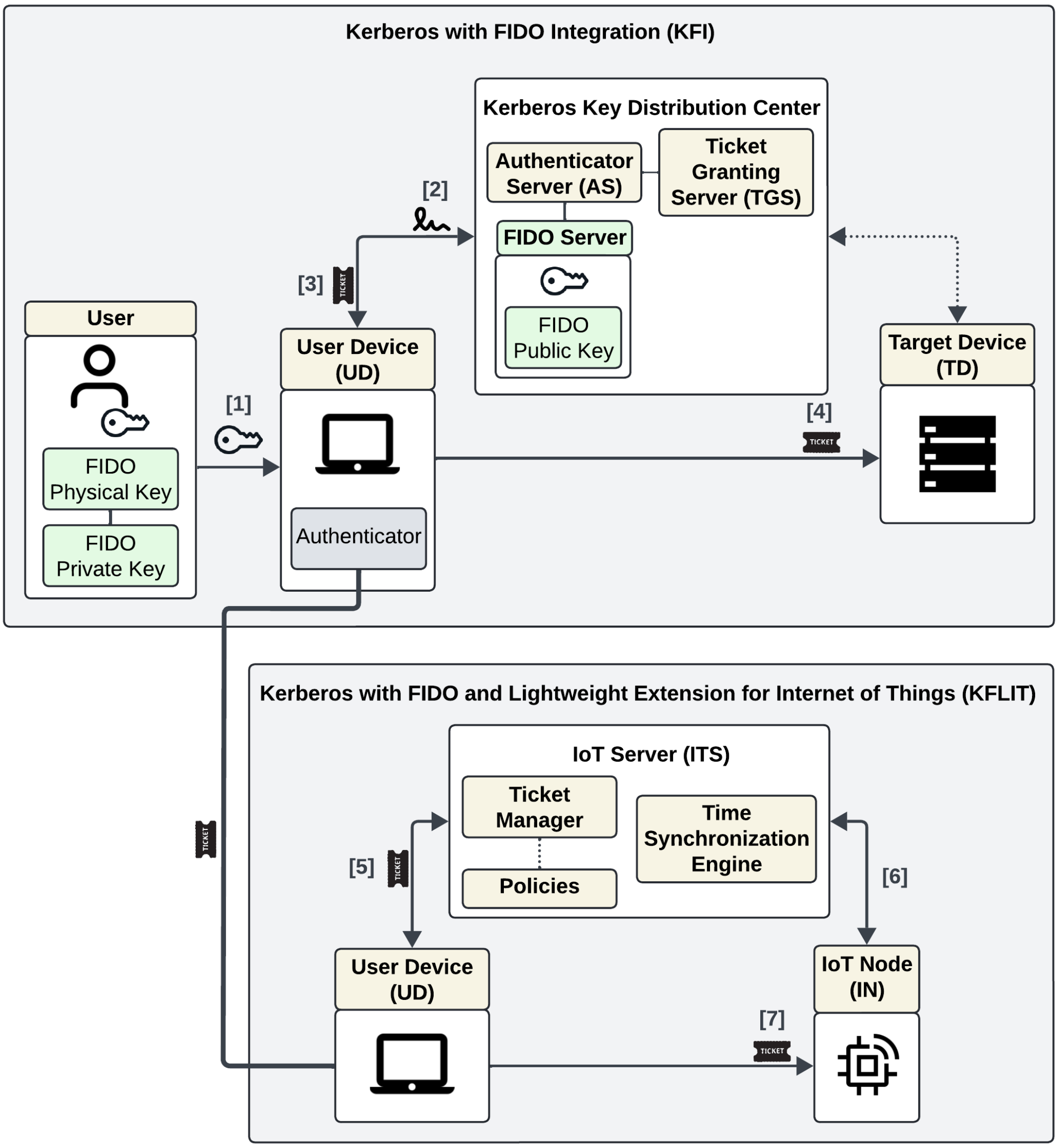


Figure 4.1: KFI and KFLIT system model.

4.4 System Model

This thesis proposes two novel protocols designed to overcome the limitations of traditional Kerberos while extending its applicability to address modern authentication challenges. Fig. 4.1 illustrates the overall architecture, which depicts both protocols: *KFI* and *KFLIT*.

1. ***Kerberos with FIDO Integration (KFI):*** *KFI* eliminates password-based vulnerabilities in traditional Kerberos by integrating FIDO's passkey mechanism. This integration replaces static passwords with a secure, passwordless authentication framework based on public-key cryptography. The entities involved are:

   (a) ***User:*** Possesses a physical FIDO key that securely stores the FIDO private key ($FIDO\text{-}Priv_{UD}$) used for authentication. The user interacts with the user device (UD) by inserting the FIDO key and completing the authentication challenge.

(b) ***User Device*** (UD)***:*** Initiates the authentication process to access a protected target device (TD). It is equipped with a FIDO authenticator that signs challenges using the private key stored in the physical FIDO key. UD communicates with the Kerberos key distribution center (KDC) to request and obtain tickets.

(c) ***Kerberos Key Distribution Center*** (KDC)***:*** A trusted authentication entity comprising two components:

→ *Authenticator Server (AS):* Verifies the FIDO-signed challenge during initial authentication and issues a ticket-granting ticket (TGT).

→ *Ticket Granting Server (TGS):* Receives TGT and issues service-specific tickets to allow access to TD.

(d) ***Target Device*** (TD)***:*** Represents the resource or application that UD intends to access securely. TD validates the service ticket and mutual authenticator before granting access.

2. ***Kerberos with FIDO and Lightweight Extension for Internet of Things (KFLIT):*** *KFLIT* extends *KFI* to address constraints specific to IoT environments. It replaces traditional encryption with lightweight HMAC and XOR operations to reduce computational overhead, uses counter-based synchronization to eliminate dependency on real-time clocks, and includes an attestation mechanism to verify device integrity. *KFLIT* introduces an additional server, the IoT server (ITS), to enable secure interaction between the user device (UD) and the IoT node (IN). The entities involved in *KFLIT* are:

(a) ***User Device*** (UD)***:*** Reuses the ticket obtained via the *KFI* protocol to initiate communication with ITS. UD is the requesting device in IoT environments, and it must be proven authentic before accessing IN.

(b) ***IoT Server*** (ITS)***:*** Serves as the security coordinator in IoT environments. It validates the ticket issued by KDC, issues access tickets for IN, and ensures integrity through device attestation. ITS comprises:

→ *Ticket Manager:* Handles issuing and verifying IoT-specific service tickets.

→ *Policies Module:* Enforces predefined access control rules for secure UD-IN communication.

→ *Time Synchronization Engine:* Maintains synchronization with IN using counters, thereby avoiding reliance on real-time clocks.

(c) ***IoT Node*** (IN)***:*** A lightweight endpoint representing an IoT device that UD wants to access. IN collaborates with ITS for attestation, allowing ITS to verify that the node has not been compromised before granting access.

In the *KFI* protocol, the process starts with the user [1] inserting the FIDO physical key into UD, which performs passwordless authentication with *AS* [2] using the FIDO private-public key pair. Upon successful verification, *AS* generates TGT and interacts with TGS [3] to issue a service ticket. UD then uses this ticket to securely access TD [4]. In the *KFLIT* protocol, UD communicates with ITS [5] using the ticket acquired from the *KFI* phase. ITS performs ticket validation, synchronizes with IN [6] using counter-based methods, and verifies device integrity through attestation. Upon successful verification, UD receives permission to interact with IN [7], completing a lightweight, secure authentication tailored for IoT environments.

Together, *KFI* and *KFLIT* provide a unified and secure authentication framework. While *KFI* enhances traditional Kerberos by integrating FIDO-based passwordless authentication, *KFLIT* further optimizes it for scalable and resource-efficient authentication in IoT ecosystems.

4.5 Adversary Model

The adversary model is defined under the following assumptions:

1. The adversary can launch remote attacks targeting IN to compromise its software and manipulate its behaviour. However, the adversary cannot compromise the hardware integrity or access securely stored credentials within IN.
2. The adversary cannot compromise the software, data, or cryptographic keys residing within the trusted servers, including AS, TGS, and ITS. These servers are assumed to be fully secure and trusted.
3. The adversary can compromise UD through various means, including impersonation (to bypass TD/IN access policies), eavesdropping (to intercept session keys or authentication data), tampering (to alter message content and disrupt protocol flow), and replay attacks (to reuse intercepted messages for unauthorized access).
4. The adversary cannot access the FIDO private keys stored within UD or on the external FIDO security keys. These keys remain protected within secure hardware modules and are inaccessible to external adversaries.

4.6 Proposed Protocol Framework

The proposed framework comprises two protocols designed to overcome the limitations of traditional Kerberos and adapt them to modern authentication challenges: *KFI* and *KFLIT*. Table 4.2 presents the notations utilized to explain the proposed scheme.

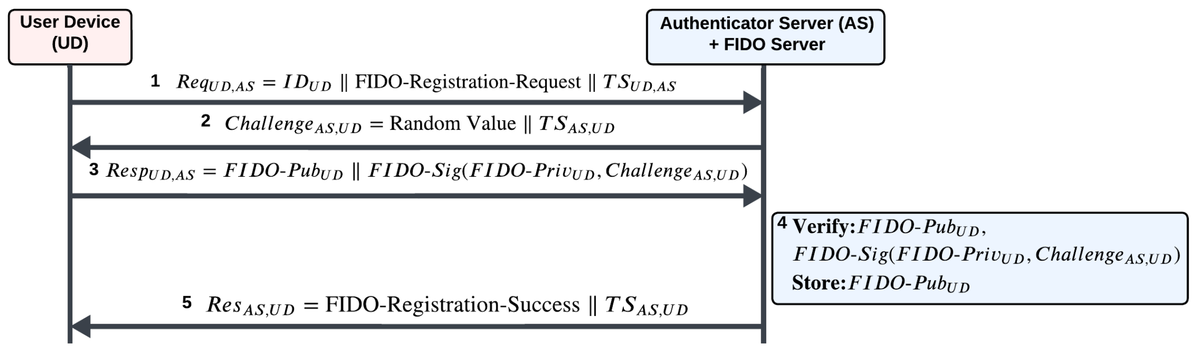


Figure 4.2: KFI: FIDO registration phase.

4.6.1 KFI: Kerberos with FIDO Integration

KFI integrates FIDO's passwordless authentication mechanism into its framework, aligning with the industry's shift toward passwordless authentication. Its main phases include registration, authentication, and ticket-granting, each detailed below.

4.6.1.1 FIDO Registration Phase

Step 1: UD initiates the FIDO registration by sending the registration request:

$$Req_{UD,AS} = ID_{UD} \parallel \text{FIDO-Registration-Request} \parallel TS_{UD,AS} \tag{4.1}$$

Step 2: AS responds by generating a random challenge and timestamp:

$$Challenge_{AS,UD} = \text{Random Value} \parallel TS_{AS,UD} \tag{4.2}$$

Step 3: UD signs the challenge using its FIDO private key and sends the following response:

$$Resp_{UD,AS} = FIDO\text{-}Pub_{UD} \parallel FIDO\text{-}Sig(FIDO\text{-}Priv_{UD}, Challenge_{AS,UD}) \tag{4.3}$$

Step 4: AS verifies the received signature using the public key and, upon successful verification, stores the key:

$$\text{Verify: } (FIDO\text{-}Pub_{UD}, FIDO\text{-}Sig(FIDO\text{-}Priv_{UD}, Challenge_{AS,UD}))$$

$$\text{Store: } FIDO\text{-}Pub_{UD}$$

Step 5: AS sends a registration success confirmation:

$$Res_{AS,UD} = \text{FIDO-Registration-Success} \parallel TS_{AS,UD} \tag{4.4}$$

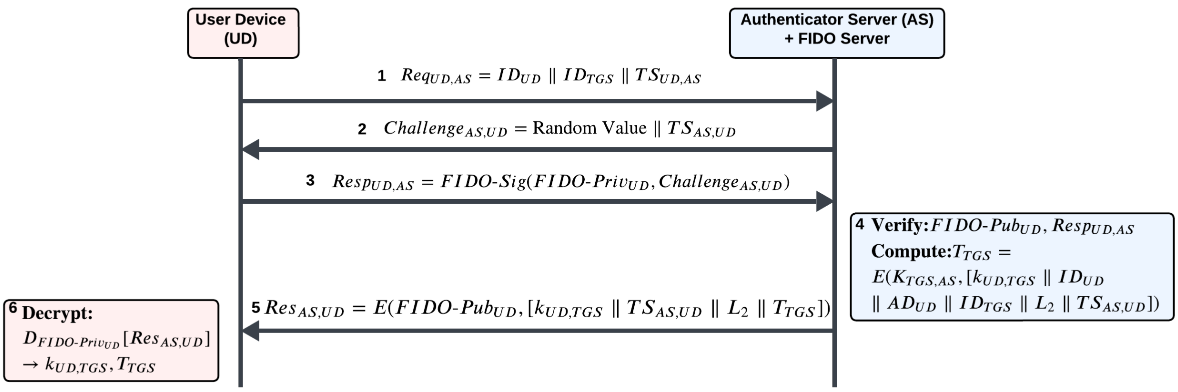


Figure 4.3: KFI: Initial authentication with AS.

4.6.1.2 Initial Authentication with AS

Step 1: UD initiates the authentication by computing the request $Req_{UD,AS}$ (Eq. 4.5), which contains its identity, the identity of the TGS, and a timestamp:

$$Req_{UD,AS} = ID_{UD} \parallel ID_{TGS} \parallel TS_{UD,AS} \tag{4.5}$$

Step 2: AS responds with a challenge $Challenge_{AS,UD}$ (Eq. 4.6) that includes a random value and timestamp:

$$Challenge_{AS,UD} = \text{Random Value} \parallel TS_{AS,UD} \tag{4.6}$$

Step 3: UD computes the response $Resp_{UD,AS}$ by signing the received challenge using its FIDO private key:

$$Resp_{UD,AS} = FIDO\text{-}Sig(FIDO\text{-}Priv_{UD}, Challenge_{AS,UD}) \quad (4.7)$$

Step 4: AS verifies the FIDO signature and generates the TGS ticket (Eq. 4.8). It then encrypts the response using UD's FIDO public key (Eq. 4.9):

$$T_{TGS} = E_{K_{TGS,AS}}[k_{UD,TGS} \parallel ID_{UD} \parallel AD_{UD} \parallel ID_{TGS} \parallel L_2 \parallel TS_{AS,UD}] \quad (4.8)$$

Step 5: AS sends $Res_{AS,UD}$ to UD.

$$Res_{AS,UD} = E_{FIDO\text{-}Pub_{UD}}[k_{UD,TGS} \parallel TS_{AS,UD} \parallel L_2 \parallel T_{TGS}] \quad (4.9)$$

Step 6: UD decrypts the response and extracts $k_{UD,TGS}$ and T_{TGS}.

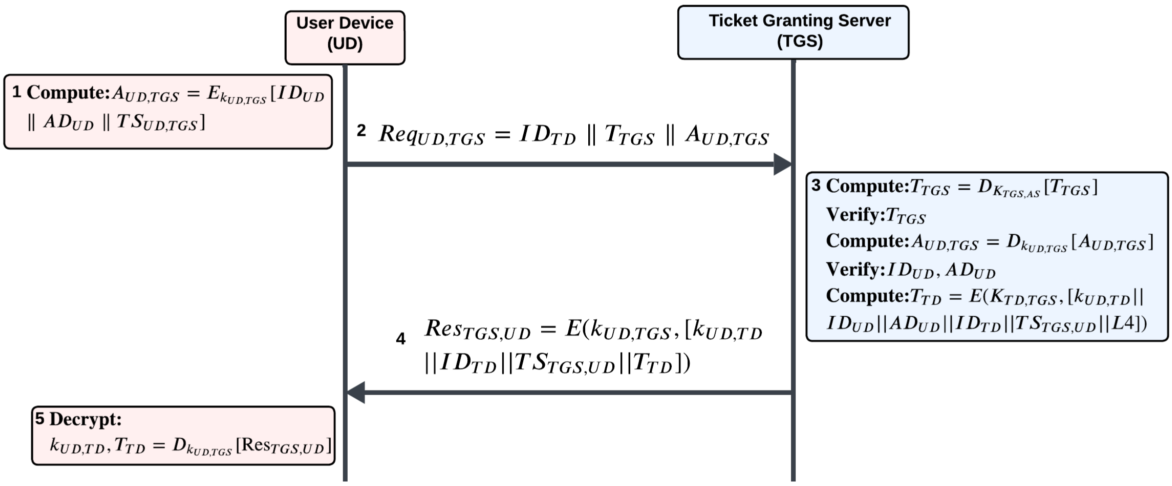


Figure 4.4: KFI: TGS interaction phase.

4.6.1.3 TGS Interaction Phase

Step 1: UD computes the authenticator $A_{UD,TGS}$ using its session key shared with the TGS and constructs the request:

$$A_{UD,TGS} = E_{k_{UD,TGS}}[ID_{UD} \parallel AD_{UD} \parallel TS_{UD,TGS}] \quad (4.10)$$

Step 2: UD sends $Req_{UD,TGS}$ to TGS.

$$Req_{UD,TGS} = ID_{TD} \parallel T_{TGS} \parallel A_{UD,TGS} \quad (4.11)$$

Step 3: TGS performs the following operations:

→ Decrypts T_{TGS} using $K_{TGS,AS}$

→ Verifies the contents of T_{TGS}

→ Decrypts $A_{UD,TGS}$ using $k_{UD,TGS}$

→ Verifies the identity and address of UD

→ Computes the ticket T_{TD} and response $Res_{TGS,UD}$

$$T_{TD} = E_{K_{TD,TGS}}[k_{UD,TD} \parallel ID_{UD} \parallel AD_{UD} \parallel ID_{TD} \parallel TS_{TGS,UD} \parallel L_4] \tag{4.12}$$

Step 4: TGS sends $Res_{TGS,UD}$ to UD.

$$Res_{TGS,UD} = E_{k_{UD,TGS}}[k_{UD,TD} \parallel ID_{TD} \parallel TS_{TGS,UD} \parallel T_{TD}] \tag{4.13}$$

Step 5: UD decrypts $Res_{TGS,UD}$ using $k_{UD,TGS}$ and retrieves:

$$k_{UD,TD}, T_{TD} = D_{k_{UD,TGS}}[Res_{TGS,UD}]$$

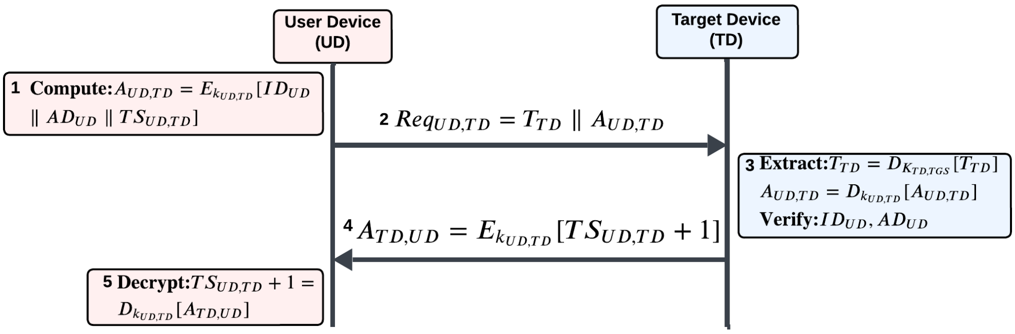


Figure 4.5: KFI: Accessing the service.

4.6.1.4 Accessing the Service

Step 1: UD computes the authenticator $A_{UD,TD}$ using its session key shared with the target device and constructs the request:

$$A_{UD,TD} = E_{k_{UD,TD}}[ID_{UD} \parallel AD_{UD} \parallel TS_{UD,TD}] \tag{4.14}$$

Step 2: UD sends $Req_{UD,TD}$ to TD.

$$Req_{UD,TD} = T_{TD} \parallel A_{UD,TD} \tag{4.15}$$

Step 3: TD performs the following actions:

→ Decrypts T_{TD} using $K_{TD,TGS}$

→ Decrypts $A_{UD,TD}$ using $k_{UD,TD}$

→ Verifies ID_{UD} and AD_{UD}

Step 4: TD computes the response and sends it back to UD.

$$A_{TD,UD} = E_{k_{UD,TD}}[TS_{UD,TD} + 1] \tag{4.16}$$

Step 5: UD decrypts the received message using $k_{UD,TD}$ and verifies it to ensure the freshness and authenticity of the session.

$$TS_{UD,TD} + 1 = D_{k_{UD,TD}}[A_{TD,UD}]$$

4.6.2 KFLIT: Kerberos with FIDO and Lightweight Extension for Internet of Things

KFLIT extends the fourth phase of *KFI*, leveraging its robust passwordless authentication mechanism as a foundational component. By adapting Kerberos for IoT, *KFLIT* addresses challenges in resource-constrained and clockless devices. Enhancing the lightweight model of KESIC [17], *KFLIT* eliminates encryption operations, replacing them with computationally efficient HMAC and XOR mechanisms. It has an attestation phase to verify device integrity before granting access, ensuring trust in authentication. This design optimizes Kerberos for IoT-specific requirements, providing a scalable, efficient, and secure authentication framework while integrating the security strengths of *KFI*.

4.6.2.1 Attestation and Time Synchronization Phase

The Attestation and Time Synchronization Phase ensures the integrity of IN and synchronizes its counter with ITS. The steps involved are as follows:

Step 1: IN increments its counter and computes:

$$A_{IN,ITS} = HMAC(K_{ITS,IN}, [ID_{IN} \parallel CO_{\text{sync}}]) \tag{4.17}$$

Step 2: IN sends the attestation request to ITS:

$$Req_{IN,ITS} = ID_{IN} \parallel CO_{\text{sync}} \parallel A_{IN,ITS} \tag{4.18}$$

Step 3: ITS verifies the counter value, and HMAC then generates a challenge and computes:

$$A^{\text{attest}}_{ITS,IN} = HMAC(K_{IN,ITS}, [ID_{ITS} \parallel \text{Challenge}]) \tag{4.19}$$

Step 4: ITS sends the attestation challenge to IN:

$$Req^{\text{attest}}_{ITS,IN} = ID_{ITS} \parallel \text{Challenge} \parallel A^{\text{attest}}_{ITS,IN} \tag{4.20}$$

Step 5: IN verifies the received attestation HMAC and computes:

$$k_{\text{attest},IN,ITS} = HMAC(K_{IN,ITS}, [\text{Challenge}]) \tag{4.21}$$

$$\text{Attst}_{\text{HMAC}} = HMAC(k_{\text{attest},IN,ITS}, [\text{Memory}]) \tag{4.22}$$

Step 6: IN sends the attestation response:

$$\text{Attst}_{\text{Response},IN,ITS} = \text{Attst}_{\text{HMAC}} \tag{4.23}$$

Step 7: ITS verifies the attestation response and computes the following:

$$A_{ITS,IN} = HMAC(K_{ITS,IN}, [ID_{ITS} \parallel CO_{\text{sync}} \parallel TS_{ITS,IN}]) \tag{4.24}$$

Step 8: ITS sends the time synchronization response to IN:

$$Res_{ITS,IN} = ID_{ITS} \parallel CO_{\text{sync}} \parallel TS_{ITS,IN} \parallel A_{ITS,IN} \tag{4.25}$$

Step 9: IN verifies the authenticity of the response and updates its local time reference.

$$\text{Verify: } A_{ITS,IN} = HMAC(K_{ITS,IN}, [ID_{ITS} \parallel CO_{\text{sync}} \parallel TS_{ITS,IN}])$$

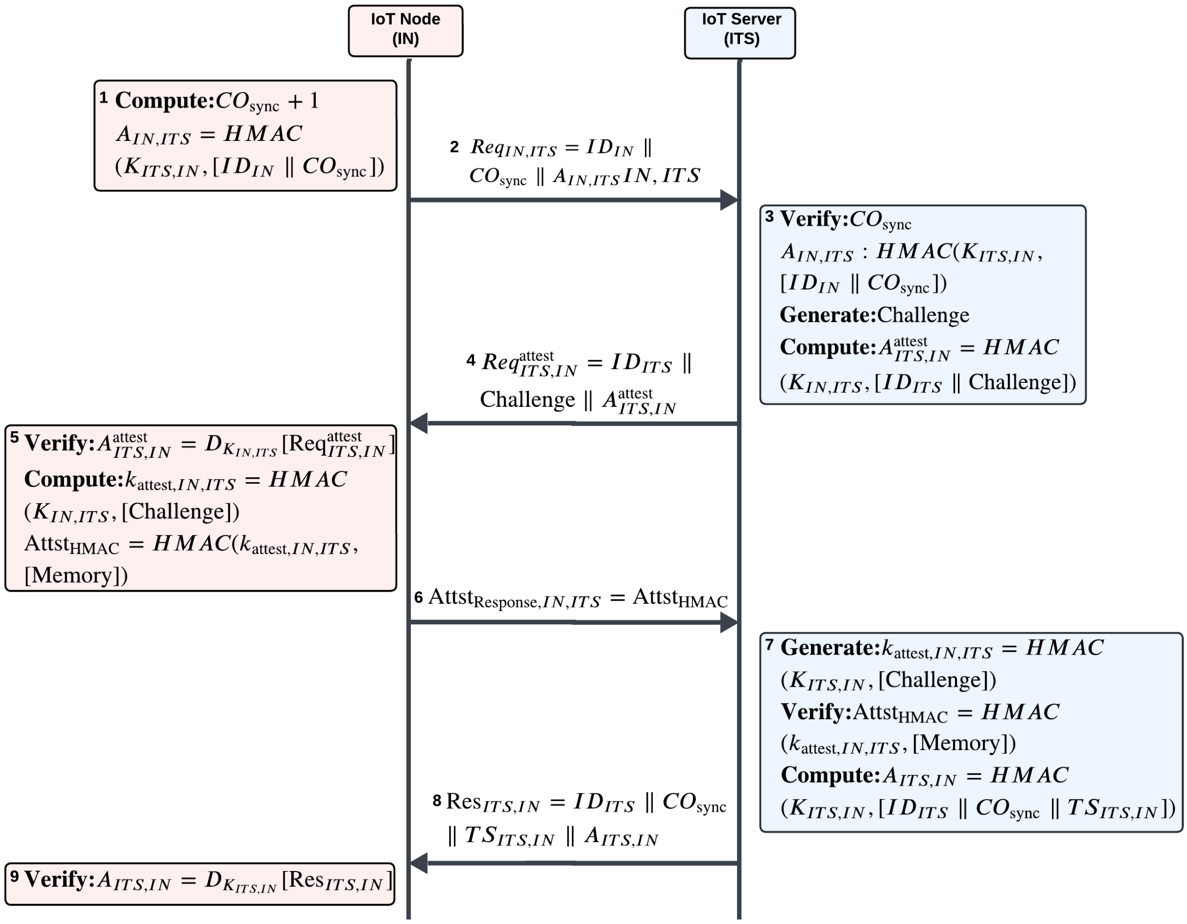


Figure 4.6: KFLIT: Attestation and time synchronization phase.

4.6.2.2 Ticket Issuing Phase

The Ticket Issuing Phase enables ITS to securely issue tickets for IN by interacting with UD. This phase relies on TGT for ITS, T_{ITS}, issued through the *KFI* protocol. By leveraging T_{ITS}, ITS ensures that only authorized UDs can request tickets for accessing IoT Nodes. The steps involved are as follows:

Step 1: UD computes the HMAC-based authenticator:

$$A_{UD,ITS} = HMAC(k_{UD,ITS}, [ID_{UD} \parallel AD_{UD} \parallel TS_{UD,ITS}]) \tag{4.26}$$

Step 2: UD sends the request to ITS:

$$Req_{UD,ITS} = ID_{IN} \parallel T_{ITS} \parallel A_{UD,ITS} \parallel TS_{UD,ITS} \tag{4.27}$$

Step 3: ITS decrypts and verifies the received ticket and computes:

$$T_{IN} = HMAC(K_{IN,ITS}, [ID_{UD} \parallel AD_{UD} \parallel L_6 \parallel ID_{IN} \parallel TS_{ITS,UD}]) \tag{4.28}$$

$$Mask = [ID_{IN} \parallel k_{UD,IN} \parallel TS_{ITS,UD} \parallel L_6 \parallel T_{IN}] \oplus k_{UD,ITS} \tag{4.29}$$

Step 4: ITS responds with an obfuscated payload and authentication code:

$$Res_{ITS,UD} = [Mask \parallel HMAC(k_{UD,ITS}, Mask)] \oplus k_{UD,ITS} \tag{4.30}$$

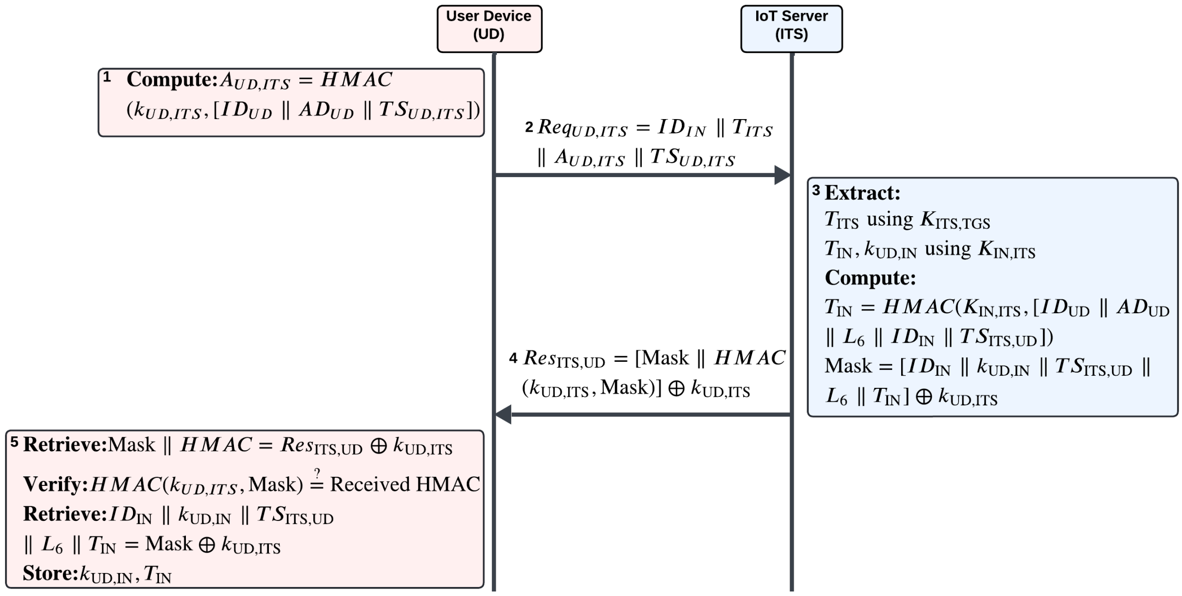


Figure 4.7: KFLIT: Ticket issuing phase.

Step 5: UD retrieves and verifies the response:

$$\text{Retrieve: } Mask \parallel HMAC = Res_{ITS,UD} \oplus k_{UD,ITS} \tag{4.31}$$

$$\text{Verify: } HMAC(k_{UD,ITS}, Mask) \stackrel{?}{=} \text{Received HMAC} \tag{4.32}$$

$$\text{Retrieve: } ID_{IN} \parallel k_{UD,IN} \parallel TS_{ITS,UD} \parallel L_6 \parallel T_{IN} = Mask \oplus k_{UD,ITS} \tag{4.33}$$

$$\text{Store: } k_{UD,IN}, T_{IN} \tag{4.34}$$

This secure mechanism ensures UD receives credentials to access IN while maintaining confidentiality and integrity through HMAC and XOR-based masking.

4.6.2.3 Service Phase

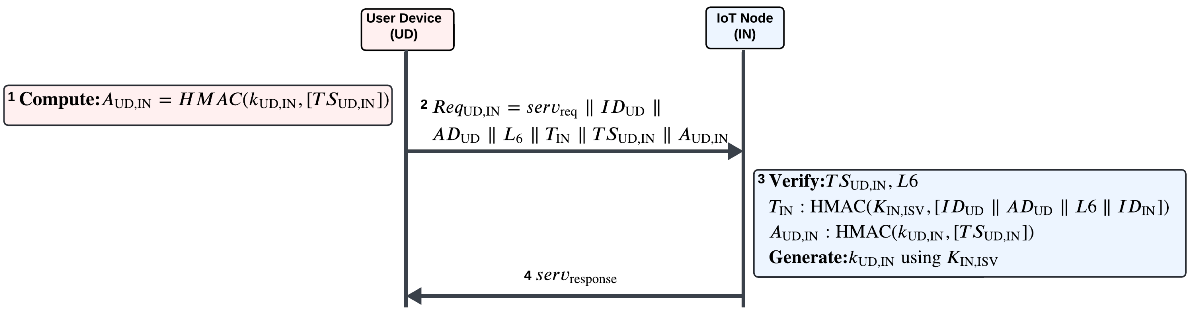


Figure 4.8: KFLIT: Service phase.

The Service Phase ensures secure communication between UD and IN for accessing requested services. The steps involved are as follows:

Step 1: UD computes the authenticator to initiate the service request:

$$A_{UD,IN} = HMAC(k_{UD,IN}, [TS_{UD,IN}]) \tag{4.35}$$

Step 2: UD sends the service request to IN:

$$Req_{UD,IN} = serv_{req} \parallel ID_{UD} \parallel AD_{UD} \parallel L_6 \parallel T_{IN} \parallel TS_{UD,IN} \parallel A_{UD,IN} \tag{4.36}$$

Step 3: IN verifies the received timestamp and level value, then validates:

$$T_{IN} = HMAC(K_{IN,ISV}, [ID_{UD} \parallel AD_{UD} \parallel L_6 \parallel ID_{IN}]) \tag{4.37}$$

$$A_{UD,IN} = HMAC(k_{UD,IN}, [TS_{UD,IN}]) \tag{4.38}$$

If the verifications are successful, IN generates $k_{UD,IN}$ using $K_{IN,ISV}$ if not already derived.

Step 4: IN responds with $serv_{response}$, completing the service access phase.

Chapter 5

Results

This chapter presents the comprehensive evaluation of the proposed authentication protocols—P-MASFEP and KFLIT—designed to enhance the security and efficiency of IoT environments. The review encompasses qualitative and quantitative analyses, including informal security assessments, computation and communication cost comparisons, and formal Scyther tool verification. P-MASFEP is tailored for IoMT applications, addressing threats such as offline password guessing and privileged insider attacks. At the same time, KFLIT builds upon Kerberos and FIDO principles to provide lightweight, scalable authentication for general IoT ecosystems. The results demonstrate that both protocols balance security robustness and performance efficiency, making them suitable for deployment in resource-constrained and security-critical IoT domains.

5.1 Security and Performance Evaluation of P-MASFEP

5.1.1 Informal Security Analysis

The informal security analysis demonstrates that the P-MASFEP fulfils the security prerequisites of IoMT-based healthcare services.

1. ***Resistant to privileged insider attacks:*** SN is responsible for generating the session key SK and compute $SN_4 = E_{(PubK_{UD})}(SK)$. SN_4 is shared with UD via GW. Even if a privileged insider at GW manages to steal TID^0_{UD}, $PubK_{\text{UD}}$, D_{ID}, and D_{LN}, they cannot retrieve SK without the private key of UD, $PvtK_{\text{UD}}$, which is only known to UD. Therefore, the P-MASFEP protocol is secure against privileged insider attacks.

2. ***Resistant to offline password guessing attacks:*** The proposed P-MASFEP scheme strengthens defence against offline password guessing by employing a fuzzy extractor with biometric information BIO_{i} to compute $FE.Gen(BIO_{\text{i}}) = (K, h_d)$ and $CPW_{\text{i}} = h(K \| D_{\text{ID}} \| D_{\text{LN}} \| TID^0_{\text{UD}})$. Since the user's biometrics are not stored for authentication, an attacker cannot guess the password. Additionally, CPW_{i} is dependent on TID_{UD}, which updates with each session. Therefore, the P-MASFEP scheme enhances defence against offline password-guessing attacks.

3. ***Resilient against replay attacks:*** Suppose an adversary captures the message $\{GW_1, GW_2, GW_3\}$ and later attempts to relay it to UD. Upon receiving the message $\{GW_1, GW_2, GW_3\}$, UD verifies the freshness of the nonce N^{1*}_{GW}. UD terminates the session since the relayed message contains an outdated nonce. This

procedure is applied to all messages, ensuring that P-MASFEP is secure against replay attacks.

4. ***Secure against DOS attacks:*** The hash and cryptographic encryption using asymmetric keys restrict the adversary's ability to obtain the session key, making it extremely difficult for an adversary to decipher the message. The entities verify the integrity of each received message. Therefore, a DoS attack is impractical in P-MASFEP.

5. ***Resistant to MITM attacks:*** Imagine that the message $\{UD_2,\ UD_3\ \ SN^{*}_{\mathrm{IEI}}\}$ was intercepted by an adversary. Any try to perform MITM will be unsuccessful since the messages $UD_2 = h(C^{0*}_{\mathrm{UD}} \| N^{1*}_{\mathrm{GW}} \| R^{0*}_{\mathrm{UD}} \| TID^{0}_{\mathrm{UD}})$, $UD_3 = N^{2}_{\mathrm{UD}} \oplus D_{\mathrm{LN}}$ and $SN^{*}_{\mathrm{IEI}} = h(D_{\mathrm{ID}} \| D_{\mathrm{LN}} \| R^{0*}_{\mathrm{UD}} \| N^{1*}_{\mathrm{GW}}) \oplus SN_{\mathrm{IEI}}$ are computed through bitwise XOR and cryptographic hash function. The collision-resistant property of hash functions [35] makes it harder for the adversary to retrieve or predict the values. Hence, P-MASFEP is secure against MITM attacks.

6. ***Resistant to impersonation attacks:*** Suppose an adversary intercepts the message $\{GW_5,\ GW_6,\ GW_7,\ PubK^{*}_{\mathrm{UD}}\}$. Due to the collision-resistant property of hash functions [35], it is computationally infeasible for the attacker to retrieve $PubK_{\mathrm{UD}}$ or TID^{0}_{UD}. Additionally, each device is equipped with a distinct PUF, making it impossible for the attacker to replicate its identity.

7. ***Protection against physical attacks:*** If an adversary physically tampers with SN to clone or extract data from its chip, PUFs in UD and SN act as safeguards. Any interference with PUF would damage the unique properties of UD, rendering it useless since PUF output depends on inherent physical variations in the integrated circuit [33]. Thus, P-MASFEP is protected from cloning and side-channel attacks.

8. ***Exhibits message freshness and integrity:*** The P-MASFEP scheme sends the messages as a message digest and checks the freshness of the shared messages over the channel using nonces. These techniques ensure that the received message remains unaltered during transmission, thereby verifying the integrity of the data.

9. ***Mutual authentication:*** UD, GW, and SN perform mutual authentication and verify the genuineness before establishing a session key to secure further communication. The effective execution of this protocol is contingent upon the legitimacy of every entity involved in the authentication process.

10. ***Ensures data privacy:*** Suppose an adversary captures the message $\{GW_9,\ PubK^{*}_{\mathrm{SN}},\ GW_{10},\ C^{1*}_{\mathrm{UD}},\ SN_4,\ SN_5\}$ in an attempt to extract sensitive information. In this message, N^{3}_{GW} and D_{ID} are enclosed as $GW_9 = N^{3}_{\mathrm{GW}} \oplus D_{\mathrm{ID}}$. The actual values of D_{ID} and N^{3}_{GW} are never transmitted over the communication channel. As a result, the adversary is unable to retrieve this information. Moreover, other parameters are computed using bitwise XOR and cryptographic hash operations, ensuring the privacy of the message's contents.

11. ***Session key security:*** The real SK is never disclosed over the public channel; instead, it is encrypted as $SN_4 = E_{(PubK_{UD})}(SK)$, which ensures secrecy over the public channel. As a result, an adversary cannot retrieve SK. Thus, the P-MASFEP scheme ensures session key security.

Table 5.1: Computation cost calculations of P-MASFEP & related schemes.

Scheme	**Computation Cost**
Alladi et al. [9]	$8C_{\text{HMAC}} + 16C_{\text{F}}$
Gope et al. [21]	$22C_{\text{H}} + 3C_{\text{HMAC}}$
Shao et al. [22]	$37C_{\text{H}}$
MASK [11]	$23C_{\text{H}} + 4C_{\text{F}}$
P-MASFEP	$28C_{\text{H}} + 2C_{\text{AED}} + C_{\text{FE}}$

C_{H}: Computation of Hash Function, C_{HMAC}: Computation of Hash Message Authentication Code, C_{F}: Computation of Cryptographic Function, C_{AED}: Computation of Asymmetric Key Encryption/Decryption, C_{FE}: Computation of Fuzzy Extractor.

12. ***User device and sensor node identity anonymity and untraceability:*** Suppose an adversary intercepts the message $\{N_{\text{UD}}^{1*}, TID_{\text{UD}}^{0*}\}$ in an attempt to extract real D_{LN} and D_{ID}. The adversary cannot retrieve this information, as D_{LN} and D_{ID} are never used during the session key establishment phase. Additionally, GW generates new temporary identities, TID_{UD} and TID_{SN}, for future communication and converts these temporary identities into pseudo-identities during mutual authentication and key establishment. Moreover, the scheme updates the temporary identities each session, preventing an adversary from tracking UD or SN.

5.1.2 Comparative Analysis

This section compares P-MASFEP to evaluate the scheme's functionality, communication, and computation costs against those of other relevant schemes.

Computation Cost Comparison

The computational overhead encompasses all cryptographic operations to establish mutual authentication and session keys. Multiprecision Integer and Rational Arithmetic Library (MIRACL) [51] is an open-source library designed to facilitate the testing of cryptographic protocols. It provides a robust platform for performing complex arithmetic operations on large integers and rational numbers, essential in various cryptographic applications, such as public key cryptography. This paper uses the methodology outlined by Yu et al. [36] based on MIRACL to derive the computation costs. The following assumptions are taken into account while calculating the computation costs of various schemes: C_{H}: Computation of Hash Function $\approx$ 0.309 ms (for example, Secure Hash Algorithm (SHA-256) [52]), C_{HMAC}: Computation of Hash Message Authentication Code (HMAC) $\approx$ 0.618 ms (HMAC takes approximately twice the time of a standalone hash operation [53]), C_{F}: Computation of Cryptographic Function $\approx$ 0.012 ms (for example, Advanced Encryption Standard (AES) [54]), C_{AED}: Computation of Asymmetric Key Encryption/Decryption $\approx$ 0.522 ms (for example, Rivest Shamir Adelman (RSA) [55]) and C_{FE}: Computation of Fuzzy Extractor $\approx$ 2.848 ms. It is worth noting that the total computational cost of the proposed P-MASFEP scheme is 10.690 ms, which is slightly more than MASK's [11] 7.155 ms. Table 5.1 outlines each protocol's detailed computation cost calculations, highlighting the operations involved in the cost estimation.

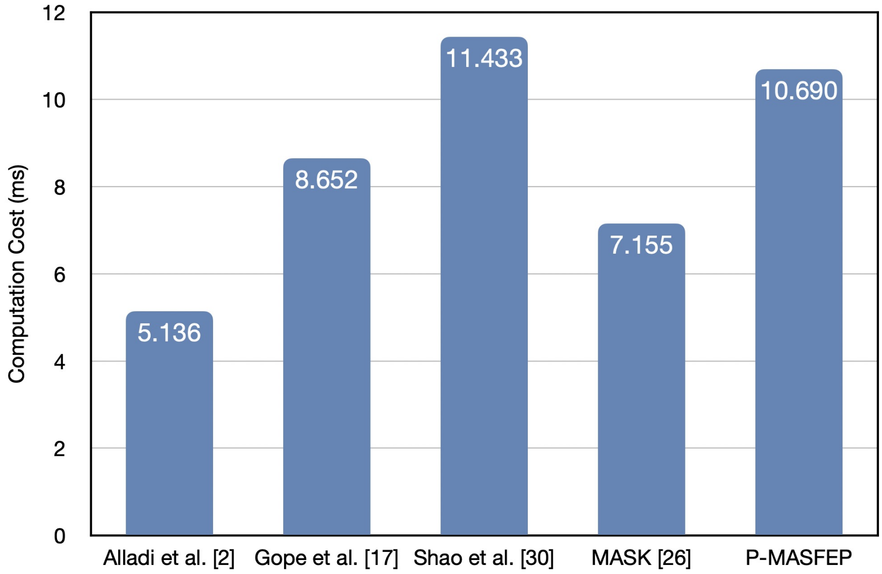


Figure 5.1: Computation cost comparison of P-MASFEP.

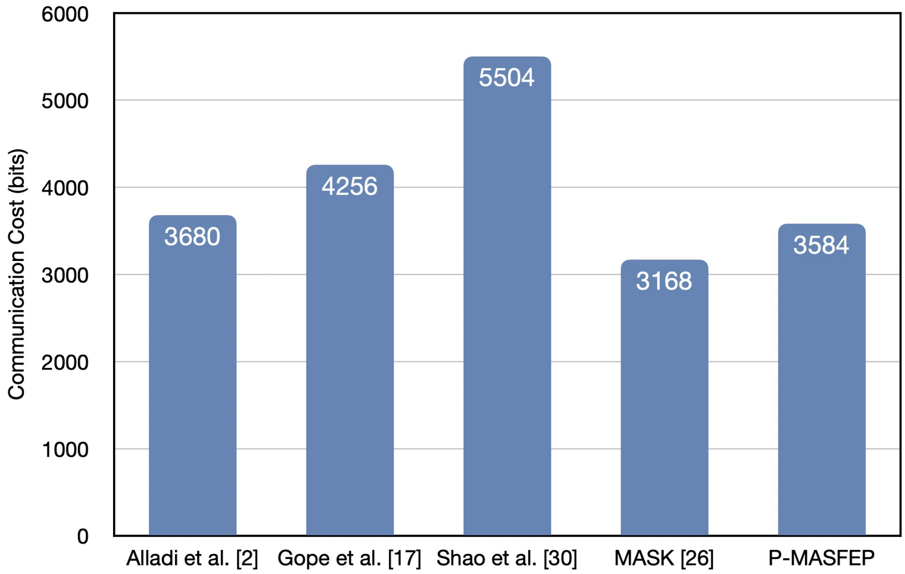


Figure 5.2: Communication cost comparison of P-MASFEP.

Fig. 5.1 visually compares the total computational costs of P-MASFEP and other schemes. Although P-MASFEP incurs a slightly higher computational overhead, its robust security guarantees justify this trade-off, which is critical for IoMT applications. Compared to different schemes, P-MASFEP achieves a favourable balance between computational efficiency and security, making it a practical solution for resource-constrained IoMT devices.

Communication Cost Comparison

The communication overhead is the amount of information the participants send or receive to complete the authentication process. The following assumptions are taken into account while calculating the communication costs of various schemes: the hash is 160 bits, random nonces, and the identities are 128 bits, the cryptographic encryption/decryption block is 256 bits, PUF is 128 bits, and the timestamp is 32 bits according to Banerjee et al. [34]. Fig. 5.2 visually represents the communication cost comparison for various schemes. The MASK [11] scheme has a communication cost of 3168 bits, whereas the proposed P-MASFEP scheme has a communication cost of 3584 bits, indicating a marginal increase in overhead. Although MASK [11] incurs a lower communication cost, it is insecure as it is vulnerable to offline password guessing and privileged insider attacks. In contrast, the slight increase in P-MASFEP's communication cost is justified by its enhanced security guarantees, making it a robust choice for IoMT systems.

5.1.3 Formal Verification Using Scyther

Scyther [29] is a widely used automated protocol verification tool. Scyther can verify and characterize protocols, producing a finite representation of all potential behaviour. It validates multiple security properties using the Security Protocol Description Language (SPDL) script. The script describes protocols as a collection of roles containing events and different claims to reflect desired security features like secrecy, non-injunctive synchronization (Nisynch), aliveness of roles (Aliveness), non-injunctive agreement (Niagree) and weak agreement of roles (Weakagree) [30]. The script verifies seventeen secret claim events to confirm the scheme's confidentiality, security, and legitimacy. Additionally, Niagree, Alive, Nisynch, and Weakagree authentication claims are evaluated for each of the three roles to ensure the overall security of the proposed P-MASFEP scheme. According to the Scyther simulation, P-MASFEP satisfies all security and authentication standards. The script is available in Annexure I. Fig. 5.3 shows the simulation output used to evaluate P-MASFEP.

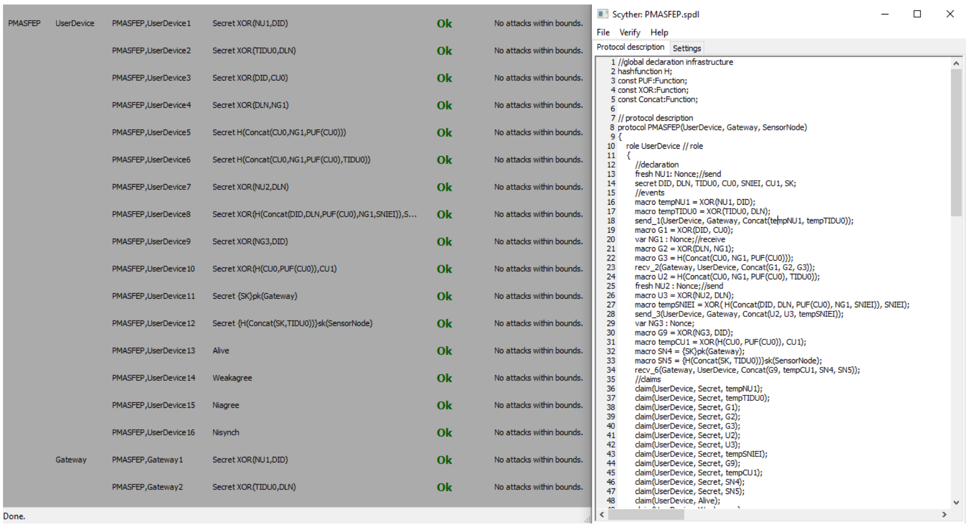


Figure 5.3: Result obtained from Scyther for the P-MASFEP scheme.

Table 5.2: Security comparison of KESIC & KFLIT.

Attack Type	**KESIC [17]**	**KFLIT (Proposed)**
Password-Guessing and Dictionary Attacks	×	✓
Kerberoasting	×	✓
Golden Ticket Attacks	×	✓
Silver Ticket Attacks	×	✓
Replay Attacks	✓	✓
Man-in-the-Middle (MitM) Attacks	✓	✓

✓: Attack Mitigated   ×: Attack Not Mitigated

5.2 Security and Performance Evaluation of KFLIT

5.2.1 Informal Security Analysis

To assess the resilience of the proposed protocols against modern attack vectors, Table 5.2 presents a comparative analysis of *KFLIT* and *KESIC* [17]. It highlights the key security improvements introduced in *KFLIT*, demonstrating its effectiveness in mitigating a broader range of attacks, including those not addressed by existing schemes like *KESIC*.

1. ***Password-Guessing and Dictionary Attacks:*** Using public-private key cryptography, *KFI* eliminates static passwords, rendering brute-force and dictionary attacks infeasible. *KFLIT* inherits this passwordless foundation, ensuring secure authentication specifically adapted for resource-constrained IoT environments.

2. ***Kerberoasting:*** Kerberoasting attacks [18] exploit weak encryption in Kerberos tickets to extract service account credentials for offline brute-force attacks. *KFI*

mitigates this threat by eliminating password-derived keys and using FIDO's asymmetric authentication mechanism, making it impractical for attackers to extract usable credentials from tickets.

3. ***Golden Ticket Attacks:*** Golden Ticket attack [19] involves forging TGTs to obtain unrestricted access. *KFI* ensures that TGTs cannot be forged without access to the FIDO private key, which remains securely stored on the user's physical key and is never transmitted. *KFLIT* carries forward this security to IoT environments, preserving integrity even across lightweight nodes.

4. ***Silver Ticket Attacks:*** Silver Ticket attacks [20] aim to impersonate users at the service level by forging service-specific tickets. In *KFI*, FIDO's challenge-response process ensures client authenticity, thwarting impersonation attempts. *KFLIT* strengthens this by using HMAC-bound session tickets, preventing ticket forgery and reuse, and maintaining secure communication with IoT nodes.

5. ***Replay Attacks:*** Replay attacks involve reusing captured authentication messages to gain unauthorized access. *KFI* neutralizes this threat by generating unique FIDO-signed challenge responses per session, ensuring message freshness. *KFLIT* complements this mechanism with counter-based synchronization, removing the reliance on real-time clocks while effectively detecting and blocking replayed messages.

6. ***Man-in-the-Middle (MitM) Attacks:*** MitM attacks aim to intercept and manipulate communication between parties. *KFI* prevents such attacks by enforcing message integrity through FIDO. *KFLIT* enhances this with HMAC-secured communication between UD, ITS, and IN, ensuring tamper-proof data transmission across enterprise and IoT settings.

5.2.2 Comparative Analysis

This section compares *KFLIT* with *KESIC* [17] to evaluate its communication and computation costs. The comparison highlights the improvements in efficiency and security introduced by *KFLIT* while extending the lightweight principles of *KESIC*.

Computation Cost Comparison

The computational overhead includes all cryptographic operations required for mutual authentication. To evaluate the operations, the Multiprecision Integer and Rational Arithmetic Library (MIRACL) [51] is used. It relies on following computational cost assumptions [36]: C_{HMAC}: 0.618 ms, $C_{\text{Encryption/Decryption}}$: 0.572 ms, and C_{XOR}: 0.003 ms. *KFLIT* achieves enhanced security at a lower computational cost than *KESIC*.

Communication Cost Comparison

Communication costs are calculated based on assumptions of data block sizes for IDs (8 bytes), synchronization counters and challenges (32 bytes), symmetric encryption blocks (256 bytes), XOR operations (12 bytes), HMAC blocks (64 bytes), and timestamps (4 bytes) [34]. Table 5.4 demonstrates that while *KFLIT* increases the number of messages, its communication cost is lower due to the efficient use of lightweight operations. In contrast, *KFI* incurs a higher communication cost owing to its additional FIDO-based

Table 5.3: Computation cost comparison of KFLIT.

Scheme	With Attestation (ms)	Without Attestation (ms)
Kerberos [12]	7.822	-
KFI (Proposed)	7.214	-
KESIC [17]	10.670	6.614
KFLIT (Proposed)	9.960	5.412

Table 5.4: Communication cost comparison of KFLIT.

Scheme	Number of Messages	Communication Cost (Bytes)
Kerberos [12]	7	672
KFI (Proposed)	12	814
KESIC [17]	6	696
KFLIT (Proposed)	8	584

exchanges. Still, this overhead is justified by its ability to mitigate advanced attacks such as Kerberoasting, Golden Ticket, and Silver Ticket, which are not effectively addressed in existing schemes like *KESIC*.

5.2.3 Formal Verification Using Scyther

Scyther verifies and characterizes protocols, producing a finite representation of all potential behaviours. It validates multiple security properties using the Security Protocol Description Language (SPDL) script. The script describes protocols as a collection of roles containing events and claims to reflect desired security properties such as secrecy, non-injective synchronization (Nisynch), aliveness of roles (Alive), non-injective agreement (Niagree), and weak agreement of roles (Weakagree) [30]. For the proposed protocols, Scyther confirms that *KFI* and *KFLIT* satisfy all evaluated security and authentication properties, ensuring their robustness against potential threats. Table 5.5 summarizes the verified security attributes for the protocols. The script is available in Annexure II.

Table 5.5: Verified security attributes of KFI & KFLIT.

KFI					
Security Attributes	Secrecy	Nisynch	Aliveness	Niagree	Weakagree
User Device	✓	✓	✓	✓	✓
Authenticator Server	✓	✓	✓	✓	✓
Ticket Granting Server	✓	✓	✓	✓	✓
Target Device	✓	✓	✓	✓	✓
KFLIT					
Security Attributes	Secrecy	Nisynch	Aliveness	Niagree	Weakagree
User Device	✓	✓	✓	✓	✓
IoT Server	✓	✓	✓	✓	✓
IoT Node	✓	✓	✓	✓	✓

✓: Verified.

Chapter 6

Conclusion and Future Work

Secure, efficient, and scalable authentication mechanisms are essential for the evolving landscape of IoT. The two novel protocols, P-MASFEP and KFLIT, address distinct but complementary challenges in IoT authentication through innovative cryptographic and architectural enhancements. P-MASFEP focuses on IoMT, where security vulnerabilities such as offline password-guessing and privileged insider attacks pose significant risks to patient data confidentiality and system integrity. By integrating PUFs, FE, and PKI, P-MASFEP ensures secure mutual authentication and session key establishment. KFLIT extends the capabilities of traditional Kerberos by incorporating FIDO's passwordless authentication model and tailoring the protocol for IoT ecosystems. The foundational scheme, KFI, addresses password-derived threats such as Kerberoasting, Golden Ticket, and Silver Ticket attacks by replacing password-based login with FIDO's cryptographic passkeys. Building on KFI, KFLIT introduces lightweight cryptographic operations (HMAC and XOR), counter-based synchronization instead of timestamping, and attestation mechanisms to verify IoT device integrity. These enhancements significantly reduce computational complexity while maintaining Kerberos' robust ticket-based framework, making KFLIT well-suited for secure and efficient authentication in large-scale and heterogeneous IoT deployments. Future work will explore the practical deployment of both protocols in real-world environments. For P-MASFEP, this includes addressing deployment challenges across various healthcare infrastructures. For KFI and KFLIT, future directions include performance benchmarking in high-traffic networks, enhancing support for cross-domain authentication in distributed IoT environments, and ensuring interoperability across heterogeneous infrastructures.

Annexure I

Scyther Script for P-MASFEP

```
//global declaration infrastructure
hashfunction H;
const PUF:Function;
const XOR:Function;
const Concat:Function;
// protocol description
protocol PMASFEP(UserDevice, Gateway, SensorNode)
{
    role UserDevice // role
    {
        //declaration
        fresh NU1: Nonce;//send
        secret DID, DLN, TIDU0, CU0, SNIEI, CU1, SK;
        //events
        macro tempNU1 = XOR(NU1, DID);
        macro tempTIDU0 = XOR(TIDU0, DLN);
        send_1(UserDevice, Gateway, Concat(tempNU1, tempTIDU0));
        macro G1 = XOR(DID, CU0);
        var NG1 : Nonce;//receive
        macro G2 = XOR(DLN, NG1);
        macro G3 = H(Concat(CU0, NG1, PUF(CU0)));
        recv_2(Gateway, UserDevice, Concat(G1, G2, G3));
        macro U2 = H(Concat(CU0, NG1, PUF(CU0), TIDU0));
        fresh NU2 : Nonce;//send
        macro U3 = XOR(NU2, DLN);
        macro tempSNIEI = XOR( H(Concat(DID, DLN, PUF(CU0), NG1, SNIEI)), SNIEI);
        send_3(UserDevice, Gateway, Concat(U2, U3, tempSNIEI));
        var NG3 : Nonce;
        macro G9 = XOR(NG3, DID);
        macro tempCU1 = XOR(H(CU0, PUF(CU0)), CU1);
        macro SN4 = {SK}pk(Gateway);
        macro SN5 = {H(Concat(SK, TIDU0))}sk(SensorNode);
        recv_6(Gateway, UserDevice, Concat(G9, tempCU1, SN4, SN5));
        //claims
        claim(UserDevice, Secret, tempNU1);
        claim(UserDevice, Secret, tempTIDU0);
        claim(UserDevice, Secret, G1);
        claim(UserDevice, Secret, G2);
        claim(UserDevice, Secret, G3);
        claim(UserDevice, Secret, U2);
        claim(UserDevice, Secret, U3);
        claim(UserDevice, Secret, tempSNIEI);
        claim(UserDevice, Secret, G9);
        claim(UserDevice, Secret, tempCU1);
```

```
    claim(UserDevice, Secret, SN4);
    claim(UserDevice, Secret, SN5);
    claim(UserDevice, Alive);
    claim(UserDevice, Weakagree);
    claim(UserDevice, Niagree);
    claim(UserDevice, Nisynch);
}
role Gateway // role
{
    //declaration
    var NU1: Nonce;//receive
    secret DID, DLN, TIDU0, CU0, SNIEI, CSN0, TIDSN0, CSN1, SK, CU1;
    //events
    macro tempNU1 = XOR(NU1, DID);
    macro tempTIDU0 = XOR(TIDU0, DLN);
    recv_1(UserDevice, Gateway, Concat(tempNU1, tempTIDU0));
    macro G1 = XOR(DID, CU0);
    fresh NG1 : Nonce;//send
    macro G2 = XOR(DLN, NG1);
    macro G3 = H(Concat(CU0, NG1, PUF(CU0)));
    send_2(Gateway, UserDevice, Concat(G1, G2, G3));
    macro U2 = H(Concat(CU0, NG1, PUF(CU0), TIDU0));
    var NU2 : Nonce;//send
    macro U3 = XOR(NU2, DLN);
    macro tempSNIEI = XOR( H(Concat(DID, DLN, PUF(CU0), NG1, SNIEI)), SNIEI);
    recv_3(UserDevice, Gateway, Concat(U2, U3, tempSNIEI));
    macro G5 = XOR(SNIEI, CSN0);
    fresh NG2 : Nonce;// send
    macro G6 = XOR(TIDSN0, NG2);
    macro G7 = H(Concat(CSN0, NG2, PUF(CSN0)));
    macro tempCSN1 = XOR(H(Concat(CSN0, PUF(CSN0))), CSN1);
    send_4(Gateway, SensorNode, Concat(G5, G6, G7, tempCSN1));
    var NSN1 : Nonce; //receive
    macro SN2 = XOR(NSN1, TIDSN0);
    macro SN4 = {SK}pk(Gateway);
    macro SN5 = {H(Concat(SK, TIDU0))}sk(SensorNode);
    recv_5(SensorNode, Gateway, Concat(SN2, SN4, SN5));
    fresh NG3 : Nonce;
    macro G9 = XOR(NG3, DID);
    macro tempCU1 = XOR(H(CU0, PUF(CU0)), CU1);
    send_6(Gateway, UserDevice, Concat(G9, tempCU1, SN4, SN5));
    //claims
    claim(Gateway, Secret, tempNU1);
    claim(Gateway, Secret, tempTIDU0);
    claim(Gateway, Secret, G1);
    claim(Gateway, Secret, G2);
    claim(Gateway, Secret, G3);
    claim(Gateway, Secret, U2);
```

```
        claim(Gateway, Secret, U3);
        claim(Gateway, Secret, tempSNIEI);
        claim(Gateway, Secret, G5);
        claim(Gateway, Secret, G6);
        claim(Gateway, Secret, G7);
        claim(Gateway, Secret, tempCSN1);
        claim(Gateway, Secret, SN2);
        claim(Gateway, Secret, SN4);
        claim(Gateway, Secret, SN5);
        claim(Gateway, Secret, G9);
        claim(Gateway, Secret, tempCU1);
        claim(Gateway, Alive);
        claim(Gateway, Weakagree);
        claim(Gateway, Niagree);
        claim(Gateway, Nisynch);
    }
    role SensorNode //role
    {
        //declaration
        secret CSN0, SNIEI, TIDSN0, TIDU0, CSN1, SK;
        //events
        macro G5 = XOR(SNIEI, CSN0);
        fresh NG2 : Nonce;// send
        macro G6 = XOR(TIDSN0, NG2);
        macro G7 = H(Concat(CSN0, NG2, PUF(CSN0)));
        macro tempCSN1 = XOR(H(Concat(CSN0, PUF(CSN0))), CSN1);
        recv_4(Gateway, SensorNode, Concat(G5, G6, G7, tempCSN1));
        fresh NSN1 : Nonce;
        macro SN2 = XOR(NSN1, TIDSN0);
        macro SN4 = {SK}pk(Gateway);
        macro SN5 = {H(Concat(SK, TIDU0))}sk(SensorNode);
        send_5(SensorNode, Gateway, Concat(SN2, SN4, SN5));
        //claims
        claim(SensorNode, Secret, G5);
        claim(SensorNode, Secret, G6);
        claim(SensorNode, Secret, G7);
        claim(SensorNode, Secret, tempCSN1);
        claim(SensorNode, Secret, SN2);
        claim(SensorNode, Secret, SN4);
        claim(SensorNode, Secret, SN5);
        claim(SensorNode, Alive);
        claim(SensorNode, Weakagree);
        claim(SensorNode, Niagree);
        claim(SensorNode, Nisynch);
    }
}
```

Annexure II

Scyther Script for KFLIT

```
// Global Declarations
hash function H;          // Cryptographic Hash Function
const XOR:Function;      // XOR Function
const Concat:Function;   // Concatenation Operator
const FIDO:Function;     // FIDO-Related Operations
const TS:Function;       // Timestamp Function
const Ticket:Function;   // Kerberos Ticket Representation
protocol KFI(UD, AS, TGS, TD)
{
    role UD
    {
        fresh NU1: Nonce; // Nonce for FIDO Authentication
        secret ID_UD, FIDO_PrivUD, k_ud_tgs, k_ud_td;
        // Step 1: FIDO Registration
        send_1(UD, AS, Concat(ID_UD, "FIDO-Registration", TS(UD -> AS)));
        // Step 2: FIDO Authentication (Challenge-Response)
        recv_2(AS, UD, Concat("Challenge(AS -> UD)"));
        send_3(UD, AS, Concat(FIDO("Pub(UD)"), FIDO("Sig(FIDO_PrivUD,
        Challenge)")));
        // Step 3: Receive TGT from AS
        recv_4(AS, UD, Concat(Ticket(TGS), k_ud_tgs));
        // Step 4: Send TGT to TGS for Service Ticket
        send_5(UD, TGS, Concat(ID(TD), Ticket(TGS), HMAC(k_ud_tgs, [ID(UD),
        TS(UD -> TGS)])));
        // Step 5: Receive Service Ticket
        recv_6(TGS, UD, Concat(Ticket(TD), k_ud_td));
        // Step 6: Access Target Device
        send_7(UD, TD, Concat(Ticket(TD), HMAC(k_ud_td, [ID(UD), TS(UD -> TD)])));
        recv_8(TD, UD, HMAC(k_ud_td, [ID(TD), TS(TD -> UD)]));
        // Claims for UD
        claim(UD, Secret, FIDO_PrivUD);
        claim(UD, Secret, k_ud_tgs);
        claim(UD, Secret, k_ud_td);
        claim(UD, Secret, Ticket(TGS));
        claim(UD, Secret, Ticket(TD));
        claim(UD, Alive);
        claim(UD, Weakagree);
        claim(UD, Niagree);
        claim(UD, Nisynch);
    }
    role AS
    {
        secret ID_AS, k_as_tgs;
        var FIDO_PubUD: PublicKey;
```

```
        // Step 1: Handle FIDO Registration Request
        recv_1(UD, AS, Concat(ID(UD), "FIDO-Registration", TS(UD -> AS)));
        send_2(AS, UD, Concat("Challenge(AS -> UD)"));
        // Step 2: Verify FIDO Response and Issue TGT
        recv_3(UD, AS, Concat(FIDO_Pub(UD), FIDO_Sig));
        send_4(AS, UD, Concat(Ticket(TGS), k_ud_tgs));
        // Claims for AS
        claim(AS, Secret, Ticket(TGS));
        claim(AS, Secret, k_as_tgs);
        claim(AS, Alive);
        claim(AS, Weakagree);
    }
    role TGS
    {
        secret ID_TGS, k_tgs_td;
        var Ticket(TGS): Ticket;
        var k_ud_tgs: SessionKey;
        // Step 1: Handle Request for Target Device Ticket
        recv_5(UD, TGS, Concat(ID(TD), Ticket(TGS), HMAC(k_ud_tgs, [ID(UD),
        TS(UD -> TGS)])));
        send_6(TGS, UD, Concat(Ticket(TD), k_ud_td));
        // Claims for TGS
        claim(TGS, Secret, Ticket(TGS));
        claim(TGS, Secret, k_ud_td);
        claim(TGS, Alive);
        claim(TGS, Weakagree);
    }
    role TD
    {
        secret ID_TD, k_td_tgs, k_ud_td;
        var Ticket(TD): Ticket;
        // Step 1: Handle Service Request
        recv_7(UD, TD, Concat(Ticket(TD), HMAC(k_ud_td, [ID(UD), TS(UD -> TD)])));
        send_8(TD, UD, HMAC(k_ud_td, [ID(TD), TS(TD -> UD)]));
        // Claims for TD
        claim(TD, Secret, Ticket(TD));
        claim(TD, Secret, k_ud_td);
        claim(TD, Alive);
        claim(TD, Weakagree);
    }
}
protocol KFLIT(UD, ITS, IN)
{
    role UD
    {
        fresh NU1: Nonce;
        secret ID_UD, k_ud_its, k_ud_in, T(ITS), T(IN);
        // Phase 1: Attestation and Counter Synchronization
```

```
    send_1(UD, ITS, Concat(ID(UD), T(ITS), HMAC(k_ud_its, [ID(UD),
    TS(UD -> ITS)])));
    recv_2(ITS, UD, Concat(HMAC(k_ud_its, [ID(ITS), TS(ITS -> UD)])));
    // Phase 2: IoT Ticket Request
    send_3(UD, ITS, Concat(ID(IN), T(ITS), HMAC(k_ud_its, [ID(UD),
    TS(UD -> ITS)])));
    recv_4(ITS, UD, Concat(T(IN), k_ud_in));
    // Phase 3: Service Access
    send_5(UD, IN, Concat(T(IN), HMAC(k_ud_in, [ID(UD), TS(UD -> IN)])));
    recv_6(IN, UD, HMAC(k_ud_in, [ID(IN), TS(IN -> UD)]));
    // Claims for UD
    claim(UD, Secret, k_ud_its);
    claim(UD, Secret, k_ud_in);
    claim(UD, Secret, T(ITS));
    claim(UD, Secret, T(IN));
    claim(UD, Alive);
    claim(UD, Weakagree);
    claim(UD, Niagree);
    claim(UD, Nisynch);
}
role ITS
{
    secret ID_ITS, k_its_in;
    var T(ITS): Ticket;
    var k_ud_its: SessionKey;
    // Phase 1: Handle Attestation
    recv_1(UD, ITS, Concat(ID(UD), T(ITS), HMAC(k_ud_its, [ID(UD), TS(UD
    -> ITS)])));
    send_2(ITS, UD, Concat(HMAC(k_ud_its, [ID(ITS), TS(ITS
    -> UD)])));
    // Phase 2: Issue IoT Ticket
    recv_3(UD, ITS, Concat(ID(IN), T(ITS), HMAC(k_ud_its, [ID(UD),
    TS(UD -> ITS)])));
    send_4(ITS, UD, Concat(T(IN), k_ud_in));
    // Claims for ITS
    claim(ITS, Secret, k_ud_its);
    claim(ITS, Secret, k_ud_in);
    claim(ITS, Secret, T(ITS));
    claim(ITS, Alive);
    claim(ITS, Weakagree);
}
role IN
{
    secret ID_IN, k_in_its, k_ud_in;
    var T(IN): Ticket;
    // Phase 3: Handle Service Request
    recv_5(UD, IN, Concat(T(IN), HMAC(k_ud_in, [ID(UD), TS(UD -> IN)])));
    send_6(IN, UD, HMAC(k_ud_in, [ID(IN), TS(IN -> UD)]));
```

```
        // Claims for IN
        claim(IN, Secret, T(IN));
        claim(IN, Secret, k_ud_in);
        claim(IN, Alive);
        claim(IN, Weakagree);
    }
}
```

Bibliography

[1] N. K. Jha, "Internet-of-Medical-Things," in *Proceedings of the Great Lakes Symposium on VLSI 2017*, ser. GLSVLSI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 7. [Online]. Available: https://doi.org/10.1145/3060403.3066861

[2] G. Hatzivasilis, O. Soultatos, S. Ioannidis, G. Demetriou, C. Verikoukis, and C. Tsatsoulis, "Review of Security and Privacy for the Internet of Medical Things (IoMT) Resolving the Protection Concerns for the Novel Circular Economy Bioinformatics," 2019.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[4] S. M. A. Rahman, S. Ibtisum, P. Podder, and s. M. S. Hossain, "Progression and Challenges of IoT in Healthcare: A Short Review," *International Journal of Computer Applications*, vol. 185, pp. 975–8887, 10 2023.

[5] M. K. Hasan, T. M. Ghazal, R. A. Saeed, B. Pandey, H. Gohel, A. Eshmawi, S. Abdel-Khalek, and H. M. Alkhassawneh, "A Review on Security Threats, Vulnerabilities, and Counter Measures of 5G Enabled Internet-of-Medical-Things," *IET Communications*, vol. 16, no. 5, pp. 421–432, 2022.

[6] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent Advances in the Internet-of-Medical-Things (IoMT) Systems Security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 2021.

[7] M. Rahman and H. Jahankhani, "Security Vulnerabilities in Existing Security Mechanisms for IoMT and Potential Solutions for Mitigating Cyber-Attacks," *Information Security Technologies for Controlling Pandemics*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238926930

[8] Z. Ashraf, Z. Mahmood, and M. Iqbal, "Lightweight Privacy-Preserving Remote User Authentication and Key Agreement Protocol for Next-Generation IoT-Based Smart Healthcare," *Future Internet*, vol. 15, no. 12, p. 386, 2023.

[9] T. Alladi, V. Chamola *et al.*, "HARCI: A Two-Way Authentication Protocol for Three Entity Eealthcare IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 361–369, 2020.

[10] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, M. Liyanage, and P. Kumar, "Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0," *IEEE Access*, vol. 8, pp. 132 808–132 824, 2020.

[11] M. Masud, G. S. Gaba, S. Alqahtani, G. Muhammad, B. B. Gupta, P. Kumar, and A. Ghoneim, "A Lightweight and Robust Secure Key Establishment Protocol for Internet of Medical Things in COVID-19 Patients Care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 694–15 703, 2020.

[12] 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.

[13] P. Joshi, "Kerberos Security in Distributed Systems," 2022.

[14] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (v5)," Tech. Rep., 2005.

[15] F. Han, M. Alkhathami, and R. Van Schyndel, "Biometric-Kerberos Authentication Scheme for Secure Mobile Computing Services," in *2013 6th International Congress on Image and Signal Processing (CISP)*, vol. 03, 2013, pp. 1694–1698.

[16] I. Downnard, "Public-Key Cryptography Extensions into Kerberos," *IEEE Potentials*, vol. 21, no. 5, pp. 30–34, 2003.

[17] R. T. Prapty, S. Jakkamsetti, and G. Tsudik, "Kesic: Kerberos Extensions for Smart, IoT and CPS Devices," in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024, pp. 1–9.

[18] D. Demers and H. Lee, "Kerberoasting: Case Studies of an Attack on a Cryptographic Authentication Technology," *International Journal of Cybersecurity Intelligence & Cybercrime*, vol. 5, no. 2, pp. 25–39, 2022.

[19] T. Grippo and H. A. Kholidy, "Detecting Forged Kerberos Tickets in an Active Directory Environment," *arXiv preprint arXiv:2301.00044*, 2022.

[20] S. Pocarovsky, M. Koppl, M. Orgon, and A. Bohacik, "Kerberos Golden Ticket Attack," in *Proceedings of the Computational Methods in Systems and Software*. Springer, 2022, pp. 677–688.

[21] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[22] X. Shao, Y. Guo, and Y. Guo, "A PUF-based Anonymous Authentication Protocol for Wireless Medical Sensor Networks," *Wireless Networks*, vol. 28, no. 8, pp. 3753–3770, 2022.

[23] Z. Tbatou, A. Asimi, Y. Asimi, Y. Sadqi, and A. Guezzaz, "A New Mutuel Kerberos Authentication Protocol for Distributed Systems." *Int. J. Netw. Secur.*, vol. 19, no. 6, pp. 889–898, 2017.

[24] A. Kadhim F. and H. Imad Mhaibes, "A New Initial Authentication Scheme for Kerberos 5 Based on Biometric Data and Virtual Password," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 2018, pp. 280–285.

[25] N. Garg, M. Wazid, J. Singh, D. P. Singh, and A. K. Das, "Security in IoMT-driven Smart Healthcare: A Comprehensive Review and Open Challenges," *Security and Privacy*, vol. 5, no. 5, p. e235, 2022.

[26] Dhilip, "Insider Attacks: Healthcare Sector's Biggest Adversary," https://www.manageengine.com/active-directory-360/manage-and-protect-identities/identitude/blogs/insider-threats-in-healthcare.html, ManageEngine, 2023, last accessed 2024/04/09.

[27] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Advances in Cryptology - EUROCRYPT 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 523–540.

[28] M. E. Hellman, "An Overview of Public Key Cryptography," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.

[29] C. J. Cremers, "The Scyther tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper," in *International Conference on Computer Aided Verification*. Springer, 2008, pp. 414–418.

[30] C. J. F. Cremers, "Scyther: Semantics and Verification of Security Protocols," 2006.

[31] T. D. Wu, "A Real-World Analysis of Kerberos Password Security." in *Ndss*, 1999.

[32] FIDO Alliance, "FIDO Alliance - Open Authentication Standards More Secure than Passwords," https://fidoalliance.org/, accessed: 2024-11-17.

[33] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[34] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. Rodrigues, and Y. Park, "Physically Secure Lightweight Anonymous User Authentication Protocol for Internet of Things Using Physically Unclonable Functions," *IEEE Access*, vol. 7, pp. 85 627–85 644, 2019.

[35] P. Gope, J. Lee, and T. Q. Quek, "Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[36] Sungjin Yu, Ashok Kumar Das, Youngho Park, and Pascal Lorenz, "SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10 374–10 388, 2022.

[37] A. Salomaa, "Public-key Cryptography," 2013.

[38] MIT Kerberos Consortium, "Protocol tutorial," https://web.mit.edu/kerberos/krb5-latest/doc/basic/, 2023, accessed: 2025-03-12.

[39] Fortinet, "What is kerberos? kerberos authentication explained," https://www.fortinet.com/resources/cyberglossary/kerberos, 2023, accessed: 2025-03-12.

[40] Wikipedia contributors, "Kerberos (protocol)," https://en.wikipedia.org/wiki/Kerberos_(protocol), 2023, accessed: 2025-03-12.

[41] Q. Gu and P. Liu, "Denial of Service Attacks," *Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications*, vol. 3, pp. 454–468, 2007.

[42] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 474–10 498, 2021.

[43] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[44] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," Cryptology ePrint Archive, Paper 2002/059, 2002. [Online]. Available: https://eprint.iacr.org/2002/059

[45] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer, 1999, pp. 388–397.

[46] B. Preneel, "Cryptographic Hash Functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[47] Y. Kassner, "Mandiant Uncovers Threat Actors Known As FIN7," 2022, accessed: 2024-11-17. [Online]. Available: https://itnerd.blog/2022/04/06/mandiant-uncovers-threat-actors-known-as-fin7/

[48] SentinelOne, "What is Kerberoasting Attack?" 2022, accessed: 2024-11-17. [Online]. Available: https://www.sentinelone.com/cybersecurity-101/threat-intelligence/what-is-kerberoasting-attack/

[49] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "3-level Secure Kerberos Authentication for Smart Home Systems using IoT," in *2015 1st International conference on next generation computing technologies (NGCT)*. IEEE, 2015, pp. 262–268.

[50] J. Chen, H. Xiao, Y. Zheng, M. M. Hassan, M. Ianni, A. Guzzo, and G. Fortino, "DKSM: A Decentralized Kerberos Secure Service-Management Protocol for Internet of Things," *Internet of Things*, vol. 23, p. 100871, 2023.

[51] Dhilip, "MIRACL Cryptographic SDK: Multiprecision Interger and Rational Arithmetic Cryptographic Library," https://github.com/miracl/MIRACL, Github, 2024, last accessed 2024/11/21.

[52] FIPS PUB, "Secure Hash Standard," https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, NIST, 2015, last accessed 2024/08/26.

[53] Kulothungan, "Cryptography and Network," https://ebooks.inflibnet.ac.in/csp11/chapter/introduction-to-network-security/, Creative Commons, 2021, last accessed 2024/08/26.

[54] FIPS PUB, “Advanced Encryption Standard,” https://csrc.nist.gov/files/pubs/fips/197/final/docs/fips-197.pdf, NIST, 2001, last accessed 2024/08/26.

[55] S. Azad, “Practical Cryptography,” https://www.oreilly.com/library/view/practical-cryptography/9781482228892/ch08.html, Auerbach Publications, 2014, last accessed 2024/08/26.

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Secure Lightweight Authentication for Internet of Things

Total Pages 75 Name of the Scholar Harshit Tyagi

Supervisor (s)

(1) Dr. Divyashikha Sethia

(2)

(3)

Department software Engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: turnitin Similarity Index: 10%, Total Word Count: 20,251

Date: 20 May 2025

Tyagi
Candidate's Signature

Divya. May 21 2025
Signature of Supervisor(s)

harshit iot2025

MTech_Thesis_Harshit_19May[1].pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:96894106

Submission Date

May 20, 2025, 10:29 PM GMT+5:30

Download Date

May 20, 2025, 10:33 PM GMT+5:30

File Name

MTech_Thesis_Harshit_19May[1].pdf

File Size

4.3 MB

75 Pages

20,251 Words

119,585 Characters

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

May 20 2025

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

159 Not Cited or Quoted 10%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
6%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Replaced Characters
240 suspect characters on 14 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **159** Not Cited or Quoted 10%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
6% Publications
6% Submitted works (Student Papers)

May 20 202

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Internet
dspace.dtu.ac.in:8080 2%

2 Internet
www.researchgate.net <1%

3 Internet
www.mdpi.com <1%

4 Publication
Foroozan Ghosairi Darbandeh, Masoumeh Safkhani. "A New Lightweight User Au... <1%

5 Publication
"The Seventh International Conference on Safety and Security with IoT", Springer... <1%

6 Publication
Priyanka Mall, Ruhul Amin, Ashok Kumar Das, Mark T. Leung, Kim-Kwang Raymo... <1%

7 Internet
cronfa.swan.ac.uk <1%

8 Publication
Mehedi Masud, Gurjot Singh Gaba, Pardeep Kumar, Andrei Gurtov. "A user-centri... <1%

9 Internet
cdn.iiit.ac.in <1%

10 Internet
ebin.pub <1%

11	Internet	
tdr.lib.ntu.edu.tw		<1%
12	Submitted works	
Manipal University Jaipur Online on 2025-01-05		<1%
13	Internet	
scrs.in		<1%
14	Internet	
www.wacscoac.org		<1%
15	Publication	
Fengling Han, Mohammed Alkhathami, Ron Van Schyndel. "Biometric-Kerberos a...		<1%
16	Internet	
ntnuopen.ntnu.no		<1%
17	Internet	
www.hindawi.com		<1%
18	Submitted works	
University of Southampton on 2017-09-08		<1%
19	Publication	
Anurag Tiwari, Manuj Darbari. "Emerging Trends in Computer Science and Its Ap...		<1%
20	Submitted works	
Christ University on 2016-04-14		<1%
21	Publication	
Manasha Saqib, Ayaz Hassan Moon. "A Novel Lightweight Multi-factor Authentica...		<1%
22	Publication	
Shubham Kumar, Kanhaiya Kumar, Abhishek Anand, Awaneesh Kumar Yadav, Ma...		<1%
23	Submitted works	
University of Warwick on 2024-09-02		<1%
24	Submitted works	
Marymount University on 2025-03-18		<1%

25 Publication
T. Kavitha, M. K. Sandhya, V. J. Subashini, Prasidh Srikanth. "Secure Communicati... <1%

26 Submitted works
University of Portsmouth on 2025-05-07 <1%

27 Internet
franckybox.com <1%

28 Submitted works
Indian Institute of Information Technology Sri City on 2024-02-23 <1%

29 Internet
docplayer.net <1%

30 Publication
Mahdi Tahavori, Farokhlagha Moazami. "Lightweight and secure PUF-based auth... <1%

31 Submitted works
RMIT University on 2024-06-13 <1%

32 Internet
wiredspace.wits.ac.za <1%

33 Submitted works
University of Bedfordshire on 2024-12-06 <1%

34 Internet
codefinity.com <1%

35 Internet
journals.mesopotamian.press <1%

36 Internet
prr.hec.gov.pk <1%

37 Submitted works
TAFE Queensland Brisbane on 2024-11-09 <1%

38 Publication
Zain Ul Abideen, Samuel Pagliarini. "Reconfigurable Obfuscation Techniques for t... <1%

39 Internet

fastercapital.com <1%

40 Submitted works

Arts, Sciences & Technology University In Lebanon on 2024-03-08 <1%

41 Publication

Ravi Raushan Kumar Chaudhary, Kakali Chatterjee. "A Lightweight PUFbased Mul... <1%

42 Submitted works

Royal Holloway and Bedford New College on 2016-08-28 <1%

43 Publication

Tejasvi Alladi, Vinay Chamola, Naren. "HARCI: A Two-Way Authentication Protoco... <1%

44 Publication

Vanga Odelu, Ashok Kumar Das, Adrijit Goswami. "An efficient ECC-based privacy-... <1%

45 Submitted works

nith on 2021-05-03 <1%

46 Internet

www.ncasia.com <1%

47 Internet

www.slideshare.net <1%

48 Internet

www.thinkmind.org <1%

49 Publication

Ahmed Mostafa, Suk Jin Lee, Yesem Kurt Peker. "Physical Unclonable Function an... <1%

50 Submitted works

City University on 2022-10-02 <1%

51 Submitted works

RMIT University on 2024-10-03 <1%

52 Publication

Shuhua Wu, Yuefei Zhu, Qiong Pu. "A novel lightweight authentication scheme wi... <1%

53 Publication

Subhabrata Rana, Fatemeh Khoda Parast, Brett Kelly, Yang Wang, Kenneth B. Ken... <1%

54 Internet

www.theseus.fi <1%

55 Submitted works

Al Zaytoonah University on 2019-02-05 <1%

56 Submitted works

Amrita Vishwa Vidyapeetham on 2024-05-29 <1%

57 Submitted works

Babes-Bolyai University on 2024-06-18 <1%

58 Submitted works

Cardiff University on 2023-04-24 <1%

59 Publication

Carlos Diaz Motero, Juan Ramon Bermejo Higuera, Javier Bermejo Higuera, Juan A... <1%

60 Publication

Cheol Ho Jeong, Kwang Seon Ahn. "Efficient RNTS system for privacy of banking o... <1%

61 Submitted works

Higher Education Commission Pakistan on 2023-06-08 <1%

62 Submitted works

Indian Institute of Information Technology Sri City on 2024-03-19 <1%

63 Submitted works

Instituto Politecnico de Portalegre on 2024-02-09 <1%

64 Publication

Lecture Notes of the Institute for Computer Sciences Social Informatics and Telec... <1%

65 Submitted works

Middlesex University on 2013-01-18 <1%

66 Publication

Prosanta Gope, Jemin Lee, Tony Q. S. Quek. "Lightweight and Practical Anonymou... <1%

67 Submitted works

Purdue University on 2025-01-08 <1%

68 Submitted works

Staffordshire University on 2024-08-12 <1%

69 Submitted works

University of Bedfordshire on 2013-09-20 <1%

70 Submitted works

University of Greenwich on 2021-09-16 <1%

71 Submitted works

University of Maryland, University College on 2014-04-05 <1%

72 Internet

di.univ-blida.dz <1%

73 Internet

dokumen.pub <1%

74 Internet

opus.lib.uts.edu.au <1%

75 Internet

vdoc.pub <1%

76 Internet

www.cs.kent.edu <1%

77 Publication

"Chapter 300425 Hand-Based Vascular Biometrics", Springer Science and Busines... <1%

78 Publication

Aiswarya S. Nair, Sabu M. Thampi. "A location-aware physical unclonable function... <1%

79 Publication

Alaa Kadhim F., Hakeem Imad Mhaibes. "A New Initial Authentication Scheme for... <1%

80 Publication

Arun Balodi, Ambar Bajpai, Vishal Jain, Piya Kovintavewat. "Security and Privacy I... <1%

81 Publication

Assa-Agyei, Kwame. "Enhancing the Performance of Cryptographic Algorithms fo... <1%

82 Publication

Behnam Zahednejad, Chong-zhi Gao. "Addressing security requirements in indust... <1%

83 Publication

Bela Genge. "A syntactic approach for identifying multi-protocol attacks", 2009 In... <1%

84 Submitted works

Delhi Technological University on 2024-05-01 <1%

85 Submitted works

Florida International University on 2024-12-14 <1%

86 Publication

J. J. Stapleton. "Security without Obscurity - A Guide to Cryptographic Architectur... <1%

87 Publication

Junyan Guo, Ye Du, Yahang Zhang, Meihong Li. "A provably secure ECC-based acc... <1%

88 Publication

Mehedi Masud, Mamoun Alazab, Karanjeet Choudhary, Gurjot Singh Gaba. "3P-SA... <1%

89 Publication

Mohammad Abdussami, Ruhul Amin, Satyanarayana Vollala. "Provably secured li... <1%

90 Publication

Mohd Shariq, Mauro Conti, Karan Singh, Sanjeev Kumar Dwivedi, Mohammad Ab... <1%

91 Publication

Mohd Shariq, Norziana Jamil, Gopal Singh Rawat, Shehzad Ashraf Chaudhry, Meh... <1%

92 Submitted works

Naval Postgraduate School on 2007-08-13 <1%

93 Submitted works

RMIT University on 2021-03-29 <1%

94 Publication

Seyed Farhad Aghili, Hamid Mala, Pedro Peris-Lopez. "Securing Heterogeneous W... <1%

95 Submitted works

University of Hertfordshire on 2024-12-09 <1%

96 Submitted works

University of Leicester on 2025-04-15 <1%

97 Submitted works

University of North Texas on 2023-04-20 <1%

98 Publication

Vomvas, Marinos. "Enhancing Security and Privacy in Future Wireless Systems: Ad... <1%

99 Publication

Wen-Wei Li, Kuo-Hui Yeh, Yan-Sing Ji, Shi-Cho Cha. "FIDO-enabled universal authe... <1%

100 Publication

Xiong Li, Jianwei Niu, Junguo Liao, Wei Liang. "Cryptanalysis of a dynamic identity... <1%

101 Publication

Xiong, Kaiqi. "Resource optimization and security in distributed computing", Proq... <1%

102 Publication

Yueyue He, Koji Inoue. "CrowdChain: A privacy-preserving crowdfunding system ... <1%

103 Internet

bbcr.uwaterloo.ca <1%

104 Internet

hull-repository.worktribe.com <1%

105 Internet

keep.lib.asu.edu <1%

106 Internet

pure.royalholloway.ac.uk <1%

107 Internet

repositories.nust.edu.pk <1%

108 Internet

www.coursehero.com <1%

109 Internet

www.ma.rhul.ac.uk <1%

110 Internet

www.regjeringen.no <1%

111 Internet

www.techscience.com <1%

harshit iot2025

MTech_Thesis_Harshit_19May[1].pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:96894106

Submission Date

May 20, 2025, 10:29 PM GMT+5:30

Download Date

May 20, 2025, 10:33 PM GMT+5:30

File Name

MTech_Thesis_Harshit_19May[1].pdf

File Size

4.3 MB

75 Pages

20,251 Words

119,585 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

May 20 2025

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

The after-conference proceeding of the CSCT 2024 will be published in Scopus Indexed Springer Book Series "Smart Innovation, Systems and Technol

3RD CONGRESS ON SMART COMPUTING TECHNOLOGIES

CSCT 2024

Organized in In-person and Online (Hybrid Mode) by

NATIONAL INSTITUTE OF TECHNOLOGY, SIKKIM

Technically Sponsored by

SOFT COMPUTING RESEARCH SOCIETY

December 14-15, 2024

LATEST NEWS

Publication Partner

Springer

Springer Nature journals are *published* in collaboration with learned societies and institutions located around the world.

INDEXING

SCOPUS, DBLP, INSPEC, Norwegian Register for Scientific Journals and Series, SCImago, WTI Frankfurt eG, zbMATH

IMPORTANT DATES

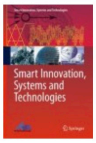
- **Last date of Full-length Submission:** October 28, 2024
- **Notification of acceptance:** November 14, 2024
- **Registration of accepted Paper:** November 29, 2024
- **Conference Date:** December 14-15, 2024

PAPER SUBMISSION

Paper submission will be through CMT using the following link

Paper Submission Link

Proceedings Publication



SCOPUS Indexed Springer Book Series, '"**Smart Innovation, Systems and Technologies**" .

Contact Us

You may send your queries to the following email ID:
csct@scrs.in
csct.scril@gmail.com
+91-7692804154 (WhatsApp messages only)

SCRS

Soft Computing Research Society ,
501, South Asian University,
Rajpur Road Maidan Garhi,
New Delhi - 110068

Gmail

23/SWE/08 HARSHIT TYAGI <harshittyagi_23swe08@dtu.ac.in>

CSCT 2024: Notification of your paper ID 728: Acceptance

Microsoft CMT <email@msr-cmt.org> Fri, 15 Nov at 00:01
Reply to: CSCT SCRS <csct.scril@gmail.com>
To: Harshit Tyagi <harshittyagi_23swe08@dtu.ac.in>

Dear Harshit Tyagi,

Thank you for submitting your research article to the CSCT 2024 - (3rd Congress on Smart Computing Technologies) to be organized by National Institute of Technology, Sikkim, India during December 14-15, 2024.

We are pleased to inform you that based on reviewers' comments, your paper titled "P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI" has been accepted for presentation during CSCT 2024, and publication in the proceedings to be published in Scopus-indexed Springer Book Series "Smart Innovation, Systems and Technologies" subject to the condition that you submit a revised version as per the comments, available at Authors CMT account. It is also required that you prepare a response to each comment from the reviewer and upload it as a separate file along with the revised paper.

The similarity index in the final paper must be less than 20%. Please note that the high plagiarism and any kind of multiple submissions of this paper to other conferences or journals will lead to rejection at any stage. Please note that the publisher, i.e. Springer Nature may ask for any other changes during the production which are supposed to be implemented. The publisher has the final right to exclude the paper from the proceedings if they found it unsuitable for publication.

Please carry out the steps to submit the camera-ready paper and online registration (Under "Regular Author" Category) as per the instructions available at

https://www.scrs.in/conference/csct2024/page/Camera_Ready_Paper_Submission

In order to register in the SCRS member category (subsidized registration fees), you can first become a member at https://www.scrs.in/register and then register for the conference OR you may register as a Regular Author Category.

Please note that the Last date for submission of the camera-ready paper, payment of the registration fee, and online registration is November 29, 2024.

Feel free to write to the "General Chairs, CSCT 2024" at csct.scril@gmail.com, should you have any questions or concerns. Please remember to always include your Paper ID- 728, whenever inquiring about your paper. For regular updates on CSCT 2024, please join the SCRS Telegram channel https://t.me/+78ZOewUxF_AwNzhl

With Regards
Team CSCT 2024 (https://www.scrs.in/conference/csct2024)
National Institute of Technology, Sikkim, India

23/SWE/08 HARSHIT TYAGI <harshittyagi_23swe08@dtu.ac.in>

CSCT 2024 (Paper ID - 728) : Registration Pending for SCOPUS Indexed Springer Book Series, ‘"SIST"

1 message

Microsoft CMT <email@msr-cmt.org> Mon, Nov 25, 2024 at 11:04 PM
Reply-To: CSCT SCRS <csct.scril@gmail.com>
To: Harshit Tyagi <harshittyagi_23swe08@dtu.ac.in>

Dear Harshit Tyagi,

Greetings!

Congratulations on the acceptance of your Paper titled "P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI” for presentation in 3rd Congress on Smart Computing Technologies (CSCT 2024) to be held on December 14-15, 2024 at National Institute of Technology, Sikkim, India and publication in the proceedings to be published in SCOPUS Indexed Springer Book Series Smart Innovation, Systems and Technologies.

As per the record, you have not registered your paper. As the last date of registration is 29 November 2024, you are requested to carry out the steps to submit the camera-ready paper and online registration as per the instructions available at

https://www.scrs.in/conference/csct2024/page/Camera_Ready_Paper_Submission

Feel free to write to the "General Chairs, CSCT 2024" at csct.scril@gmail.com, should you have any questions or concerns. Please remember to always include your Paper ID-728, whenever inquiring about your paper.

Looking forward to meeting you during the conference.

With Regards
Team CSCT 2024

Konferenza Services

Gurunanak Nagar, Gwalior,
Madhya Pradesh, India - 474012
Email: konferenza.services@gmail.com

TAX-INVOICE

Konferenza Services Website: https://konferenza.org/ GSTIN: 23ABAFK3757D1ZO	**Invoice Date**	29 November 2024
	Invoice Number	KS/NOV/2024/158

BILL ISSUED TO

Harshit Tyagi
1/4563, Street Number 3, Mandoli Road, Ramnagar Extn., Shahdara
harshittyagi_23swe08@dtu.ac.in
For Paper ID: 728
Paper Title: P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI

DESCRIPTION	
Event Name	3rd Congress on Smart Computing Technologies (CSCT2024) https://www.scrs.in/conference/csct2024
Event Dates	14 Dec 2024 - 15 Dec 2024
Category of Registration	Regular Author
Registration Fee	INR 9000
Extra Page Charges	INR 5000
GST (18%)	INR 2520
Total	**INR 16520.00** **(SIXTEEN THOUSANDS FIVE HUNDRED TWENTY RUPEES ZERO PAISE)**

Payment Transaction Id	Date Time	Mode of Payment
CSCT2024-728-113556512970	29 Nov 2024 02:13:38	CCAVENEUE

Note: Whether tax is payable under reverse charge - No

This is a computer generated invoice and needs no signature.

3rd Congress on Smart Computing Technologies

(CSCT 2024)

Organized by

National Institute of Technology, Sikkim, India

Certificate of Presentation

This certificate is proudly awarded to

Harshit Tyagi

for presenting the paper titled

P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI

authored by

Harshit Tyagi, Divyashikha Sethia

in the 3rd Congress on Smart Computing Technologies (CSCT 2024)

held during

December 14-15, 2024.

Prof. Mukesh Saraswat
General Chair

Dr. Abhishek Rajan
General Chair

Springer

https://www.scrs.in/conference/csct2024

SCRS/CSCT2024/PC/728

Licence to Publish Proceedings Papers	SPRINGER NATURE

Licensee	Springer Nature Singapore Pte Ltd.	(the 'Licensee')
Title of the Proceedings Volume/Edited Book or Conference Name:	Congress on Smart Computing Technologies *Proceedings of CSCT 2024, Volume 1*	
Volume Editor(s) Name(s):	Mukesh Saraswat, Dr. Abhishek Rajan, Dr. Antorweep Chakravorty	
Proposed Title of the Contribution:	P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI	(the 'Contribution')
Series: The Contribution may be published in the following series	Smart Innovation, Systems and Technologies	
Author(s) Full Name(s):	Harshit Tyagi, Divyashikha Sethia	(the 'Author')
When Author is more than one person the expression "Author" as used in this Agreement will apply collectively unless otherwise indicated.		
Corresponding Author Name:	Harshit Tyagi	
Instructions for Authors	https://www.springer.com/gp/authors-editors/conference-proceedings/conference-proceedings-guidelines	(the 'Instructions for Authors')

- **Grant of Rights**

  - For good and valuable consideration, the Author hereby grants to the Licensee the perpetual, exclusive, world-wide, assignable, sublicensable and unlimited right to: publish, reproduce, copy, distribute, communicate, display publicly, sell, rent and/or otherwise make available the contribution identified above, including any supplementary information and graphic elements therein (e.g. illustrations, charts, moving images) (the 'Contribution') in any language, in any versions or editions in any and all forms and/or media of expression (including without limitation in connection with any and all end-user devices), whether now known or developed in the future. Without limitation, the above grant includes: (i) the right to edit, alter, adapt, adjust and prepare derivative works; (ii) all advertising and marketing rights including without limitation in relation to social media; (iii) rights for any training, educational and/or instructional purposes; (iv) the right to add and/or remove links or combinations with other media/works; and (v) the right to create, use and/or license and/or sublicense content data or metadata of any kind in relation to the Contribution (including abstracts and summaries) without restriction. The above rights are granted in relation to the Contribution as a whole or any part and with or in relation to any other works.

- Without limiting the rights granted above, Licensee is granted the rights to use the Contribution for the purposes of analysis, testing, and development of publishing- and research-related workflows, systems, products, projects, and services; to confidentially share the Contribution with select third parties to do the same; and to retain and store the Contribution and any associated correspondence/files/forms to maintain the historical record, and to facilitate research integrity investigations. The grant of rights set forth in this clause (b) is irrevocable.

- If the Licensee elects not to publish the Contribution for any reason, all publishing rights under this Agreement as set forth in clause 1a above will revert to the Author.

- **Copyright**

Ownership of copyright in the Contribution will be vested in the name of the Author. When reproducing the Contribution or extracts from it, the Author will acknowledge and reference first publication in the Volume.

- **Use of Contribution Versions**

- For purposes of this Agreement: (i) references to the "Contribution" include all versions of the Contribution; (ii) "Submitted Manuscript" means the version of the Contribution as first submitted by the Author prior to peer review; (iii) "Accepted Manuscript" means the version of the Contribution accepted for publication, but prior to copy-editing and typesetting; and (iv) "Version of Record" means the version of the Contribution published by the Licensee, after copy-editing and typesetting. Rights to all versions of the Manuscript are granted on an exclusive basis, except for the Submitted Manuscript, to which rights are granted on a non-exclusive basis.

- The Author may make the Submitted Manuscript available at any time and under any terms (including, but not limited to, under a CC BY licence), at the Author's discretion. Once the Contribution has been published, the Author will include an acknowledgement and provide a link to the Version of Record on the publisher's website: "This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in [insert volume title], and is available online at https://doi.org/[insert DOI]".

- The Licensee grants to the Author (i) the right to make the Accepted Manuscript available on their own personal, self-maintained website immediately on acceptance, (ii) the right to make the Accepted Manuscript available for public release on any of the following twelve (12) months after first publication (the "Embargo Period"): their employer's internal website; their institutional and/or funder repositories. Accepted Manuscripts may be deposited in such repositories immediately upon acceptance, provided they are not made publicly available until after the Embargo Period.
The rights granted to the Author with respect to the Accepted Manuscript are subject to the conditions that (i) the Accepted Manuscript is not enhanced or substantially reformatted by the Author or any third party, and (ii) the Author includes on the Accepted Manuscript an acknowledgement in the following form, together with a link to the published version on the publisher's website: "This version of the contribution has been

accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/[insert DOI]. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms".
Under no circumstances may an Accepted Manuscript be shared or distributed under a Creative Commons or other form of open access licence.
Any use of the Accepted Manuscript not expressly permitted under this subclause (c) is subject to the Licensee's prior consent.

- The Licensee grants to Author the following non-exclusive rights to the Version of Record, provided that, when reproducing the Version of Record or extracts from it, the Author acknowledges and references first publication in the Volume according to current citation standards. As a minimum, the acknowledgement must state: "First published in [Volume, page number, year] by Springer Nature".

    - to reuse graphic elements created by the Author and contained in the Contribution, in presentations and other works created by them;

    - the Author and any academic institution where they work at the time may reproduce the Contribution for the purpose of course teaching (but not for inclusion in course pack material for onward sale by libraries and institutions);

    - to reuse the Version of Record or any part in a thesis written by the same Author, and to make a copy of that thesis available in a repository of the Author(s)' awarding academic institution, or other repository required by the awarding academic institution. An acknowledgement should be included in the citation: "Reproduced with permission from Springer Nature";

    - to reproduce, or to allow a third party to reproduce the Contribution, in whole or in part, in any other type of work (other than thesis) written by the Author for distribution by a publisher after an embargo period of 12 months; and

    - to publish an expanded version of their Contribution provided the expanded version (i) includes at least 30% new material (ii) includes an express statement specifying the incremental change in the expanded version (e.g., new results, better description of materials, etc.).

- **Warranties & Representations**

Author warrants and represents that:

-

    - the Author is the sole copyright owner or has been authorised by any additional copyright owner(s) to grant the rights defined in clause 1,

    - the Contribution does not infringe any intellectual property rights (including without limitation copyright, database rights or trade mark rights) or other third party rights and no licence from or payments to a third party are required to publish the Contribution,

- the Contribution has not been previously published or licensed, nor has the Author committed to licensing any version of the Contribution under a licence inconsistent with the terms of this Agreement,

- if the Contribution contains materials from other sources (e.g. illustrations, tables, text quotations), Author has obtained written permissions to the extent necessary from the copyright holder(s), to license to the Licensee the same rights as set out in clause 1 but on a non-exclusive basis and without the right to use any graphic elements on a stand-alone basis and has cited any such materials correctly;

- all of the facts contained in the Contribution are according to the current body of research true and accurate;

- nothing in the Contribution is obscene, defamatory, violates any right of privacy or publicity, infringes any other human, personal or other rights of any person or entity or is otherwise unlawful and that informed consent to publish has been obtained for any research participants;

- nothing in the Contribution infringes any duty of confidentiality owed to any third party or violates any contract, express or implied, of the Author;

- all institutional, governmental, and/or other approvals which may be required in connection with the research reflected in the Contribution have been obtained and continue in effect;

- all statements and declarations made by the Author in connection with the Contribution are true and correct;

- the signatory who has signed this Agreement has full right, power and authority to enter into this Agreement on behalf of all of the Authors; and

- the Author complies in full with: i. all instructions and policies in the Instructions for Authors, ii. the Licensee's ethics rules (available at https://www.springernature.com/gp/authors/book-authors-code-of-conduct), as may be updated by the Licensee at any time in its sole discretion.

- **Cooperation**

- The Author will cooperate fully with the Licensee in relation to any legal action that might arise from the publication of the Contribution, and the Author will give the Licensee access at reasonable times to any relevant accounts, documents and records within the power or control of the Author. The Author agrees that any Licensee affiliate through which the Licensee exercises any rights or performs any obligations under this Agreement is intended to have the benefit of and will have the right to enforce the terms of this Agreement.

- Author authorises the Licensee to take such steps as it considers necessary at its own expense in the Author's name(s) and on their behalf if the Licensee believes that a third party is infringing or is likely to infringe copyright in the Contribution including but not limited to initiating legal proceedings.

- **Author List**

  Changes of authorship, including, but not limited to, changes in the corresponding author or the sequence of authors, are not permitted after acceptance of a manuscript.

- **Post Publication Actions**

  The Author agrees that the Licensee may remove or retract the Contribution or publish a correction or other notice in relation to the Contribution if the Licensee determines that such actions are appropriate from an editorial, research integrity, or legal perspective.

- **Controlling Terms**

  The terms of this Agreement will supersede any other terms that the Author or any third party may assert apply to any version of the Contribution.

- **Governing Law**

  This Agreement shall be governed by, and shall be construed in accordance with, the laws of the Republic of Singapore. The courts of Singapore, Singapore shall have the exclusive jurisdiction.

Signed for and on behalf of the Author		Print Name:		Date:
HTyagi		Harshit Tyagi		19 January 2025

Address:		1/4563, Street Number 3, Mandoli Road, Ramanagar Extn., Shahdara, Delhi-110032, India.
Email:		harshittyagi_23swe08@dtu.ac.in

The after-conference proceeding of the ICIVC 2025 will be published in SCOPUS Indexed Springer Book Series "Lecture Notes in Networks and Systems"

ICIVC2025 Home | About ICFAI University | Camera Ready Paper Submission NEW | Call for Papers NEW | Paper Submission NEW | Registration Fee | Venue | Contact Us

ICFAI UNIVERSITY DEHRADUN

5th International Conference on Intelligent Vision and Computing (ICIVC 2025)

Organized in In-person and Online (Hybrid Mode) by

The ICFAI University, Dehradun

Technically Sponsored by

Soft Computing Research Society

June 13-14, 2025

The after-conference proceeding of the ICIVC 2025 will be published in SCOPUS Indexed Springer Book Serie

Call for Papers

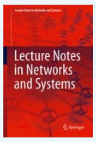
Start Date : **13 June, 2025**

End Date : **14 June, 2025**

Paper Submission Last Date : **20 April, 2025**

LATEST NEWS

Proceedings Publication



Springer Book Series "Lecture Notes in Networks and Systems"

INDEXING

SCOPUS, DBLP, INSPEC, Norwegian Register for Scientific Journals and Series, SCImago, WTI Frankfurt eG, zbMATH

PAPER SUBMISSION

Paper submission will be through CMT using the following link

Paper Submission Link

IMPORTANT DATES

- **Last date of Full-length Submission: April 20, 2025**
- **Notification of acceptance: May 15, 2025**
- **Registration of accepted Paper: May 30, 2025**
- **Conference Date: June 13-14, 2025**

Committees

- Chief Patron
- Patron
- Advisory Committee
- General Chair
- Organizing Secretary
- Organizing Chair
- Program Chair(s)
- Technical Program Committee
- Publicity Chair
- Organizing Committee
- Publication Committee
- Session Management Committee
- IT and Promotion Committee

Other Links

- Guidelines for the Appointment of Invited Speakers
- Guidelines for the selection of Session Chairs
- Guidelines for the Best Paper Selection Process
- Guidelines for Selecting and Appointing Reviewers
- Conference Rules and Guidelines
- Previous ICIVC Conferences
- Awards
- Important Dates

Contact Us

You can send your queries to the following email ID:

icivc@scrs.in

icivc2025@iudehradun.edu.in

WhatsApp Contact: +91-7692804154 (messages only)

SCRS

Soft Computing Research Society, New Delhi, India

DTU Delhi Technological University

23/SWE/08 HARSHIT TYAGI <harshittyagi_23swe08@dtu.ac.in>

ICIVC 2025: Notification of your paper ID 674: Acceptance

1 message

Microsoft CMT <noreply@msr-cmt.org> Thu, May 15, 2025 at 9:55 AM
To: Harshit Tyagi <harshittyagi_23swe08@dtu.ac.in>

Dear Harshit Tyagi,

Thank you for submitting your manuscript to 5th International Conference on Intelligent Vision and Computing (ICIVC 2025) to be held on June 13-14, 2025 at ICFAI University, Dehradun, India in Hybrid Mode. Proceedings of ICIVC 2025 will be published in the SCOPUS Indexed Springer Book Series Lecture Notes in Networks and Systems .

We are pleased to inform you that based on reviewers' comments, your paper titled "KFLIT: Kerberos with FIDO and Lightweight extension for Internet of Things" has been accepted for presentation during ICIVC 2025, and publication in the proceedings to be published in Scopus-indexed Springer Book Series "Lecture Notes in Networks and Systems" subject to the condition that you submit a revised version as per the comments, available at Authors CMT account. It is also required that you prepare a response to each comment from the reviewer and upload it as a separate file along with the revised paper.

The similarity index in the final paper must be less than 20%. Please note that the high plagiarism and any kind of multiple submissions of this paper to other conferences or journals will lead to rejection at any stage. Please note that the publisher, i.e. Springer Nature may ask for any other changes during the production which are supposed to be implemented. The publisher has the final right to exclude the paper from the proceedings if they found it unsuitable for publication.

Please carry out the steps to submit the camera-ready paper and online registration (Under "Regular Author" Category) as per the instructions available at

https://scrs.in/conference/icivc2025/page/Camera_Ready_Paper_Submission

In order to register in the SCRS member category (subsidized registration fees), you can first become a member at https://www.scrs.in/register and then register for the conference OR you may register as a Regular Author Category.

Please note that the Last date for submission of the camera-ready paper, payment of the registration fee, and online registration is May 30, 2025

Feel free to write to the "General Chairs, ICIVC 2025" at icivc.scrs@gmail.com, should you have any questions or concerns. Please remember to always include your Paper ID- 674, whenever inquiring about your paper.

With Regards
ICIVC 2025 (https://www.scrs.in/conference/icivc2025)
Join us on Telegram: https://t.me/+78ZOewUxF_AwNzhl

23/SWE/08 HARSHIT TYAGI <harshittyagi_23swe08@dtu.ac.in>

ICIVC 2025 (Paper ID - 674) : Registration Pending for Scopus-indexed Springer Book Series "LNNS"

1 message

Microsoft CMT <noreply@msr-cmt.org> Wed, May 21, 2025 at 9:19 PM
To: Harshit Tyagi <harshittyagi_23swe08@dtu.ac.in>

Dear Harshit Tyagi,

Greetings!

As per the record, you have not registered your Paper titled "KFLIT: Kerberos with FIDO and Lightweight extension for Internet of Things" for 5th International Conference on Intelligent Vision and Computing (ICIVC 2025). As the last date of registration is 30 May 2025, you are requested to carry out the steps to submit the camera-ready paper and online registration as per the instructions available at

https://scrs.in/conference/icivc2025/page/Camera_Ready_Paper_Submission

Feel free to write to the "General Chairs, ICIVC 2025" at icivc.scrs@gmail.com, should you have any questions or concerns. Please remember to always include your Paper ID-674, whenever inquiring about your paper.

Looking forward to meeting you during the conference.

With Regards
Team ICIVC 2025

Konferenza Services

Gurunanak Nagar, Gwalior,
Madhya Pradesh, India - 474012
Email: konferenza.services@gmail.com

TAX-INVOICE

Konferenza Services Website: https://konferenza.org/ GSTIN: 23ABAFK3757D1ZO	**Invoice Date**	24 May 2025
	Invoice Number	KS/MAY/2025/258

BILL ISSUED TO

Harshit Tyagi
1/4563, STREET NUMBER 3, MANDOLI ROAD, RAMNAGAR, SHAHDARA
harshittyagi_23swe08@dtu.ac.in
For Paper ID: 674
Paper Title: KFLIT: Kerberos with FIDO and Lightweight extension for Internet of Things

DESCRIPTION	
Event Name	5th International Conference on Intelligent Vision and Computing (ICIVC2025) https://scrs.in/conference/icivc2025
Event Dates	13 Jun 2025 - 14 Jun 2025
Category of Registration	Regular Author
Registration Fee	INR 10000
Extra Page Charges	INR 5500
GST (18%)	INR 2790
Total	**INR 18290.00 (EIGHTEEN THOUSANDS TWO HUNDRED NINETY RUPEES ZERO PAISE)**

Payment Transaction Id	Date Time	Mode of Payment
ICIVC2025-674-113778367022	24 May 2025 08:01:07	CCAVENEUE

Note: Whether tax is payable under reverse charge - No

This is a computer generated invoice and needs no signature.

SCRS\ICIVC2025\PC\674

5th International Conference on Intelligent Vision and Computing (ICIVC 2025)

Organized by

The ICFAI University, Dehradun

Certificate of Presentation

This certificate is awarded to

Harshit Tyagi

who has presented the paper titled **"KFLIT: Kerberos with FIDO and Lightweight extension for Internet of Things"** authored by **Harshit Tyagi, Divyashikha Sethia** in the 5th International Conference on Intelligent Vision and Computing (ICIVC 2025) held at **The ICFAI University, Dehradun** during **June 13 – 14, 2025.**

Puneet Kumar Gupta
General Chair

Springer

Apu Kumar Saha
General Chair

https://scrs.in/conference/icivc2025

DECLARATION

I hereby certify that the work which is presented in the Major Project-II entitled **Secure Lightweight Authentication for Internet of Things** in fulfillment of the requirement for the award of the Degree of Master of Technology in Software Engineering and submitted to the Department of Software Engineering, Delhi Technological University, Delhi is an authentic record of my own, carried out during a period from January to May 2025 under the supervision of **Dr. Divyashikha Sethia**.

The mater presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University. The work has been published/accepted/communicated in SCI/SCI expanded/SSCI/Scopus indexed journal OR peer reviewed Scopus indexed conference with the following details.

Title of the Paper: **P-MASFEP: security-enhanced PUF-based Mutual Authentication & Session key establishment using Fuzzy Extractor & PKI**
Author names (in sequence): Harshit Tyagi, Dr. Divyashikha Sethia
Name of Conference/Journal: 3rd Congress on Smart Computing Technologies (CSCT2024)
Conference dates with venue: 14-15th December 2024, Sikkim, India
Status of paper (Accepted/Published/Communicated): Accepted
Date of paper communication: October 10, 2024
Date of paper acceptance: November 25, 2024
Date of paper publication: July 18, 2025

Harshit Tyagi
Roll No. 23/SWE/08

Student Roll No., Name and Signature

SUPERVISOR CERTIFICATE

To the best of my knowledge, the above work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere. I further certify that the publication and indexing information given by the students is correct.

May 21 2025

Date: 20 May 2025 **Dr. Divyashikha Sethia**

Place: New Delhi **Supervisor Name and Signature**

DECLARATION

I hereby certify that the work which is presented in the Major Project-II entitled **Secure Lightweight Authentication for Internet of Things** in fulfillment of the requirement for the award of the Degree of Master of Technology in Software Engineering and submitted to the Department of Software Engineering, Delhi Technological University, Delhi is an authentic record of my own, carried out during a period from January to May 2025 under the supervision of **Dr. Divyashikha Sethia**.

The mater presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University. The work has been published/accepted/communicated in SCI/SCI expanded/SSCI/Scopus indexed journal OR peer reviewed Scopus indexed conference with the following details.

Title of the Paper: **KFLIT: Kerberos with FIDO and Lightweight extension for Internet of Things**
Author names (in sequence): Harshit Tyagi, Dr. Divyashikha Sethia
Name of Conference/Journal: 5th International Conference on Intelligent Vision and Computing (ICIVC2025)
Conference dates with venue: 13-14th June 2025, Dehradun, Uttarakhand, India
Status of paper (Accepted/Published/Communicated): Accepted
Date of paper communication: April 05, 2025
Date of paper acceptance: May 15, 2025
Date of paper publication: N/A

Tyagi

Harshit Tyagi
Roll No. 23/SWE/08

Student Roll No., Name and Signature

SUPERVISOR CERTIFICATE

To the best of my knowledge, the above work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere. I further certify that the publication and indexing information given by the students is correct.

Divya. May 21 2025

Date: 20 May 2025 **Dr. Divyashikha Sethia**

Place: New Delhi **Supervisor Name and Signature**