Dynamic Authentication Protocols for Advanced Security in Federated Metaverse Systems

by

Md Fuad Hasan
20101345
F.M. Ashfaq
24141296
Ahmed Awsaf Chowdhury
20101344
Shoeb Islam Hamim
20101337
Mustafiza Rahmani
20101006

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Department of Computer Science and Engineering
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Fuad Hasan

20101345

F.M. Ashfaq

24141296

Ahmed Awsaf Chowdhury

20101344

Shoeb Islam Hamim

20101337

Mustafiza Rahmani

20101006

Approval

The thesis/project titled "Dynamic Authentication Protocols for Advanced Security in Federated Metaverse Systems" submitted by

1. Md Fuad Hasan (20101345)
2. F.M. Ashfaq (24141296)
3. Ahmed Awsaf Chowdhury (20101344)
4. Shoeb Islam Hamim (20101337)
5. Mustafiza Rahmani (20101006)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain

Associate Professor
CSE
BRAC University

Program Coordinatior:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Metaverse is a dynamic virtual reality-based environment that is rapidly revolutionizing digital engagement. The Metaverse is propelled by the integration of technologies like blockchain, augmented reality, virtual reality, and artificial intelligence. The Metaverse is quickly increasing in acceptance as a fully immersive cyberspace for work, entertainment, and socializing. After the idea of the Metaverse, an evolution of the concept is envisioned known as the Federated Metaverse. The idea of the Federated Metaverse presents it as a digital domain that is dynamically organized as an interconnected network of separately managed worlds known as Metaworlds. These several realms function independently within the Federated Metaverse based on specific purposes, in contrast to conventional metaverse models, and are each subject to their own set of laws, regulations, and user experiences. While users traverse through these Metaworlds in this federated architecture, identity verification becomes critical to ensure appropriate permission and authentication that comply with the unique regulations and goals of each Metaworld. As the concept of this virtual universe is expected to become increasingly integrated into our daily lives, it is imperative to address the security concerns that may come up while interacting with these digitally created worlds. To counter these possible security challenges, the paper proposes a dynamic authentication model crafted for the concept of the Federated Metaverse. The goal is to design a comprehensive authentication model that can support any algorithm based on specific requirements, using multiple factors to reinforce security. The proposed model will ensure a versatile and robust authentication process. The main emphasis of the model is to bolster user privacy and security in general to ensure a solid and reliable basis in this changing digital environment all while keeping usability in mind. The user-centered secured authentication framework will prioritize usability and simplicity of interaction to enhance the overall user experience and security within the Metaverse.

Keywords: Metaverse; Federated Metaverse; virtual reality; augmented reality; digital engagement; cyberspace; identity verification; authentication model; security; user privacy; usability; interconnected network; Metaworlds

Acknowledgement

We extend our heartfelt gratitude to our Thesis Supervisor, Muhammad Iqbal Hossain, PhD, Associate Professor at BRAC University, for his unwavering support and dedication, which have been decisive in making this research worthwhile. We also express our appreciation to the Department of Computer Science and Engineering at BRAC University for providing us with the necessary resources and valuable guidance, with honourable mention and sincere thanks to Md Sadek Ferdous, PhD, Associate Professor, Department of Computer Science and Engineering, BRAC University, for his inspiration and encouragement. Additionally, we acknowledge the researchers of all the papers we referred to, as well as all individuals contributing to the field of information security, for their efforts in making technology safer and digital environments more welcoming.

Table of Contents

List of Figures

Chapter 1

Introduction

In an age manifested by relentless digital progress and unprecedented development of technology, the rise of the Metaverse serves as a testament to humanity's limitless potential for scientific advancements and technological evolution. The Metaverse, an amalgam of blockchain, artificial intelligence, and virtual and augmented reality technology has redefined how people interact in digitally created environments opening the doors to boundless possibilities. It comes as a result of the ever-existing desire of humans to create and manipulate their own territory and beyond. The Metaverse mirrors a digitally created world where people interact with each other along with other worldly or imaginative entities while transcending physical and real-time barriers. The Metaverse envisions a limitless digital world where individuals transcend physical boundaries, leap into a world of boundless potential and immerse themselves in experiences that question the very foundation of normal reality.

As the concept of the metaverse continues to advance, the conceptualization of a federated metaverse has surfaced, emphasizing a cooperative and interoperable method of creating this virtual universe. Within the Federated Metaverse, numerous virtual worlds and platforms that will be called "Metaworlds" are integrated to give users a unified and connected experience. The Federated Metaverse presents a fascinating phenomenon in the digital universe; it is akin to a large universe in which discrete regions, named Metaworlds, coexist with tailored goals, governance frameworks, and distinctive user-centered environments. In this visualized concept of the Federated Metaverse, the user experience expands into a variety of individually managed worlds created by their creators based on demanding needs, surpassing the confines of a single and centralized setting which opens doors to a much wider range of functionality and more usability with organizations or individuals creating their virtual worlds to serve specific needs. Contemplate these Metaworlds as distinct realms within the larger Metaverse, each serving particular purposes and subject to distinct laws. With this collaborative thought of the coexistence of the Metaworlds, users can move between these separate, controllable worlds as easily as they would between different planets in a galaxy of planets. Whether pursuing learning environments, entertainment-focused areas, financial workplaces, or cooperative work environments, the Federated Metaverse includes a wide range of user-based environments inside the overall virtual universe.

To ensure a secure and privacy-centric experience within the Federated Metaverse, a dynamic authentication model that can support any type of algorithm based on requirements will be implemented. This innovative approach prioritizes user privacy by allowing individuals to enter diverse Metaworlds without directly sharing any sensitive data like passwords, biometrics or personal info with the respective owners of these realms. Instead, a central server within the overarching Metaverse will act as the authentication and authorization hub allowing users based on the respective Metaworld's requirements or standards. This strategic design not only mitigates the risk of potential misuse of user data by Metaworld owners but also establishes a reliable security infrastructure. This mitigates third parties from accessing sensitive information by the avoidance of entrusting multiple organizations or individuals with sensitive information that is needed by authentication processes, the Metaverse ensures that users can seamlessly navigate between Metaworlds while maintaining control over the exchange of their personal information. The Federated Metaverse's dedication to user security and data protection is emphasized by this privacy-focused authentication scheme.

1.1 Research Problem

Governments, businesses and big organizations are already developing virtual worlds to facilitate services and commerce by rendering spatial apps around the entities that will take up space in the Metaverse which is predicted to be a thirteen trillion US dollar market to be used by an estimated 5 billion users by the year 2030 (Radcliff, 2023)[28]. With such huge finances on the line and the concept of mirroring individuals in a virtual world, Metaverse is already a focal point for malicious activities, a trend which is certain to escalate in the forthcoming future as the Metaverse moves towards its expected realization steadily. To mitigate the possibilities of malicious intent or action and reduce the overall possibilities of risk factors associated with human intervention, our research aims to verify the identity of those who step into the virtual world and limit their possibility of partaking in undesired actions whether done intentionally or unintentionally. Several concerning issues lie in the existing authentication methods and those that are expected to be implemented after the complete materialization of the Metaverse. The focal issues are:

- **Incompetence of Existing Authentication Methods:** Traditional authentication methods such as passwords, biometrics, 2FA, MFA, or the use of physical devices or cards for security all have their associated risk factors. Passwords can be weak, predictable, or susceptible to shoulder surfing, leading to unauthorized access. Biometric data, once compromised, cannot be changed, and deepfake technology poses a threat. 2FA is not protected from phishing attacks, and MFA can be inconvenient to use for the regular user, compromising its effectiveness. Additionally, social engineering attacks have the potential to circumvent the additional security layers for both 2FA and MFA. Physical devices or cards can be lost, stolen, or damaged, potentially leading to unauthorized access. Despite their effectiveness, these methods require vigilant management to mitigate their associated risks.

- **Usability of Authentication Methods:** In the Metaverse, finding the right equilibrium between security and usability is crucial. Increased complexity of authentication methods might contribute to a more secure infrastructure, but hectic authentication processes can lead to user dissatisfaction and potential adoption resistance. It is imperative to keep user comfort in mind, as the successful implementation of the Metaverse relies on its vows to make life in the virtual environment easier, reflecting in the real world and not making the experience difficult for the users. Striking the sweet spot between usability and security is as challenging to do as it is effortless to say.

- **Authentication in Dynamic Environments:** The dynamic aspect of the Metaverse is apparent in how rapidly shifting virtual worlds and circumstances are. Verification of individuals within these rapid changes presents a formidable challenge. Existing authentication techniques mostly are not adaptable to these swift changes, which could lead to false positives or even authentication failures. Therefore, adopting an authentication system that has the ability to prompt reaction and efficiently adjust to dynamic situations is a significant challenge.

- **Multiplatform Authentication:** The Metaverse is expected to be used on numerous platforms across a series of devices. As it extends its implementation across different platforms and virtual environments, establishing a universal authentication system that functions consistently across all of these platforms and virtual settings is a challenging endeavor. It is crucial to guarantee a dependable and secure authentication process across all platforms, devices, and applications associated with the Metaverse, which is a tough goal to achieve. In order to enable users to effortlessly switch between various aspects of the Metaverse while maintaining a high level of security, it is necessary to coordinate efforts to build compatible authentication protocols.

1.2 Research Objective

As this dynamic digital frontier in the form of The Metaverse unveils and reaches closer to human reach now more than ever, establishing a robust security architecture with a seamless yet functional authentication method is a ubiquitous ambition in the tech world.The research aims to develop a state-of-the-art authentication approach specifically designed to adapt to the dynamic environment of the Metaverse, have unparalleled functionality across all platforms and offer a smooth user experience all while boasting a formidable security infrastructure that addresses most if not all security-related concerns. The primary objective of this study is to propose and evaluate an innovative authentication model that is dynamic and adaptable to various algorithms, wherein the main server acts as the identity provider, furnishing identities to external worlds. The external worlds validate these identities and, based on previous authorizations, grant permissions accordingly.

The proposed authentication model incorporates a meticulous process that begins with an initial assessment of device integrity, ensuring that secured mode is enabled. Subsequently, the main server initiates a request to enter the guest server, establishing a secure and controlled access mechanism within the Metaverse.

The overarching aim is to introduce a reliable and secure authentication system that aligns with the specific demands of the Metaverse. By employing a dynamic authentication model and focusing on device integrity checks, the research strives to fortify security measures and enhance user experience within the virtual realm. This authentication model not only addresses inherent vulnerabilities in conventional trust-based systems but also sets the stage for secure and immersive digital interactions.

This research aims to reinforce the security of the Metaverse against a variety of potential security issues by using the comprehensive authentication approach via a reliable model, while also setting a new benchmark for safe and immersive digital interactions enhancing user experience. By putting equal emphasis on security and user experience first, the paper aims to build a harmonious digital environment that protects identities and sensitive data while simultaneously fostering a foundation of trust and certainty for users navigating this vast digital space.

1.3 Thesis Structure

The thesis begins by outlining the research problem, identifying gaps in current authentication methods, and providing the research objective for the purpose of setting up an investigation into the improvement of authentication processes with challenging and goal-oriented characteristics. The literature review gives a proper background of existing authentication technologies, including federated authentication and public key authentication, and presents a proper review of related works that place the research within the existing body of knowledge, highlighting innovations and limitations of current methods. In the proposed model chapter, the method of authentication is shown along with examples of where the systems can be used and their process step by step. The structure and implementation chapter centers on a hands-on implementation of that model, explaining data model and protocol flow; it also provides detailed information about architecture and operational details needed to authenticate. The proposed model in question is tested under security analysis chapter so that the authentication method may be robust with respect to its three core principles (i.e., confidentiality, integrity and availability), as well as against threats and requirements through a threat model and requirement analysis process. The conclusion and future work chapter conclude the findings, discuss the implications of the research, and suggest the directions for further work, reflecting on the study's achievements and proposing potential areas to explore further for the enhancement of authentication technologies.

Chapter 2

Literature Review

2.1 Background

2.1.1 Federated Authentication

Federated authentication such as Google or Apple ID is an authentication process that allows users to access multiple applications or systems using a single set of credentials. This scheme drastically increases user convenience and provides a seamless login experience as the user does not have to remember or manage multiple passwords or log in credentials [32]. In this authentication system shown in Fig. 2.1, the identity of a user is provided or ensured by an Identity provider(IdP). Users register to an identity providing their credentials. After registering when a user initiates login to any Service Provider(SP), the service provider uses the Identity Provider to authenticate the user. This approach of authentication provides multiple benefits in terms of both convenience and security. This eliminates the need

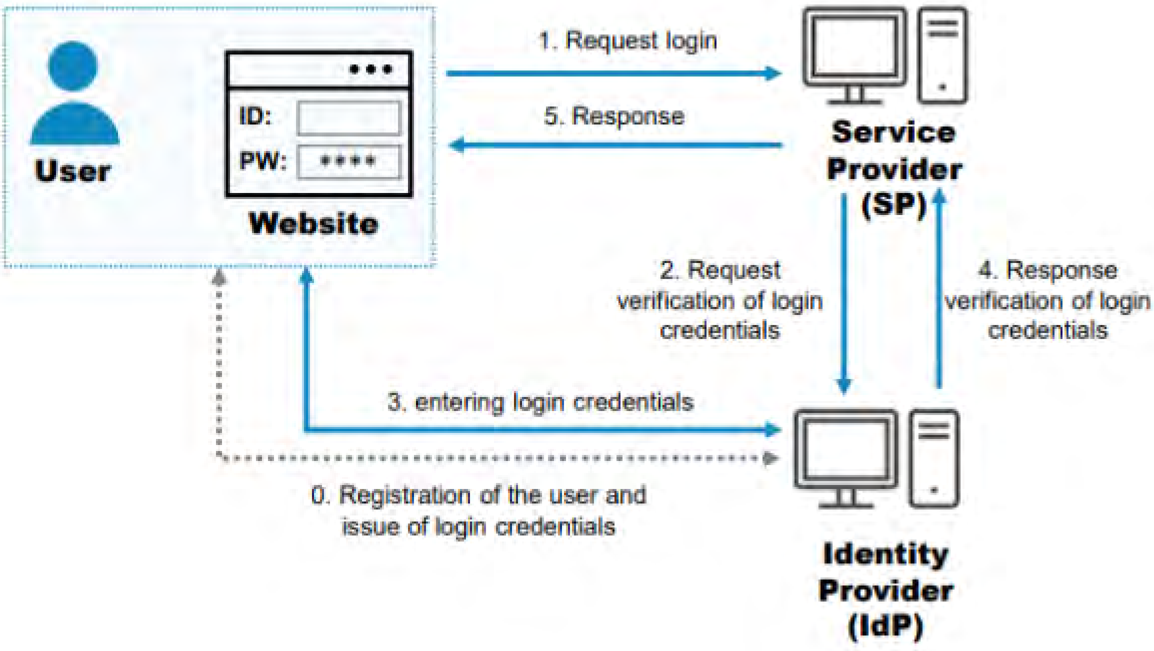

User
ID:
PW: ****
Website
1. Request login
5. Response
Service Provider (SP)
2. Request verification of login credentials
4. Response verification of login credentials
3. entering login credentials
0. Registration of the user and issue of login credentials
Identity Provider (IdP)

Figure 2.1: Federated Authentication Scheme

for managing multiple passwords and repeated logins providing credentials every time a user wants to access a service which saves time, improves the user experience, and increases productivity. Furthermore, this approach reduces password or credential-related vulnerabilities as the users do not need to share their credentials with multiple applications. However, there are some challenges and issues with this scheme. In some implementations of this approach, there is a single point of failure that means If the identity provider server fails, the user can not access other applications. Additionally, the protection of login credentials is most crucial here, if the login credentials are exposed, then user impersonation attacks can occur on multiple websites [32].

2.1.2 Public Key Authentication(PKA)

Public key authentication uses the concept of asymmetric cryptography to authenticate a user without the need to share or have a password altogether. We humans are not good at remembering things like passwords or making very creative passwords. For decades passwords had many vulnerabilities including phishing, social engineering, theft, weakness, etc. This method of authentication eliminates the vulnerabilities related to having a password.
This approach uses a public key and a private key to complete the authentication. This key pair is generated upon registration on a site and the site only knows the public key and the private key is securely stored on the device [27]. When the user tries to log in to that particular site the site creates a cryptographic challenge(using an algorithm like RSA)using the public key of that user which can only be solved using the private key and private key. The user then responds with the solved challenge which allows the user to log in without any password. This way there is no sensitive information shared between the user and the server [7].
Public key authentication provides strong security as it relies on asymmetric key pairs and the private key is kept securely stored on the device which never leaves the device. It not only eliminates password-based vulnerabilities but also protects against credential theft and management burdens.

2.2 Related Works

The paper titled “Visualization and Cybersecurity in the Metaverse: A Survey” [19] focuses on cybersecurity issues related to the visual aspect of the Metaverse along with the security threats of technologies dedicated to visualizing virtual environments. The study provides insight into the already prevailing security issues and software used to exploit them. One such is an automated user location tracking system called ARSpy for multiuser AR applications that can accurately track a victim solely based on the victim’s network traffic information. The study also gives insight into alternative authentication methods that can replace traditional methods which include RubikAuth, behavioral biometrics, user identification from kinesiological movement and AI-driven security features. RubikAuth is a 3D authentication method used in VR that verifies user identity by making them select color–digit combinations from the cube using either eye gazing, head pose, or tapping with a controller. Behavioral patterns and kinesiological movement biometrics can be combined with machine learning and AI for authenticity while within virtual environments. Additionally, probable threats are mentioned in the form of physical threats in XR environments by immersive attacks. The authors claim that techniques like XR forensics and memory forensics play a crucial role in investigating malicious activities and conclude that automated threat detection will halt possible attacks and continuous authentication methods will be the key to mitigating security risks within the Metaverse.

The article titled “BlockNet: Beyond reliable spatial Digital Twins to ParallelMetaverse” [22] extensively focuses on the creation of a proposed secure multidimensional data storage solution called “BlockNet” that will serve to improve the reliability and security of Digital Twins. Additionally in order to improve spatial data processing the article’s writers also propose a non-mutagenic multidimensional Hash Geocoding method that minimizes data loss while also improving the competence of information retrieval aiding in the implementation of the Metaverse. The writing states that Digital Twins technology can be integrated with Blockchain to ensure traceability, authenticity, quality and security. The paper further stresses the key complication of the creation of digital models which is defining spatial parameters, ranging from macroscopic to microscopic scales, storing and mapping them across storage structures and retrieving the information in an order by which it can be processed back into the digital structure. To address this issue the proposed “Hash Geocoding” method works to organize the storage of multidimensional geographic data, facilitating partitioning from macroscale to microscale. And to store all this information securely and extract it across multidimensions, the authors propose BlockNet is a concept based on Blockchain technology, to securely store and map spatial information. The main difference between BlockNet and traditional blockchain storage feature is that BlockNet is multidimensional and allows plurality of in and out degrees of each node thereby creating a network like structure whereas, traditional blockchain storage has a single in and out degree for each node thereby having a linear chain like structure. The encoding mechanism solves the complexity related to mapping and indexing of the storage of spatial information across the BlockNet improving efficiency of data exchange across all platforms including devices, IOTs, servers and storage spaces. The authors claim that the combination of their proposed storage method, BlockNet with their additionally proposed encoding mechanism Hash Geocoding can securely and effectively extract, store, map and retrieve spatial information which is the building block of Digital Twins that will shape the Metaverse.

The research conducted by Yang and colleagues [35] articulates the importance of traceability of avatars in virtual-physical environments. The paper introduces threats like harassment of avatars and establishes an urgent need for a traceable authentication mechanism. It addresses the challenges to track a malicious player who impersonates another player, Cumbersome algorithm requiring a lot of storage, one time authentication between devices. This study proposes to create an avatar and produce a signature to create non-disclosure information for the virtual identity. Chameleon collision signature algorithm which requires only a pair of keys was proposed to achieve authentication. Using the player’s biometric information with the chameleon key secures the traceability without using third parties to ensure two immersive authentication protocols. Lastly, using the chameleon key and biological samples consistently, building the two factor authentication framework equips the user for MIT (metaverse identity token) to disclose the IDP of malicious users. This model shows potential to avoid the key escrow problem while reducing computational costs.

Ajgar & Ajgar [16] worked on probable challenges for the Metaverse and focused primarily on the issues women may encounter while experiencing a 360-degree virtual environment. The authors reviewed how women are always the target when it comes to creating fake profiles and provided statistics showing that 90-95% of fraudulent profiles are using women's credentials. They pointed out that these things can occur in the Metaverse as well. It imposes strict rules on the Metaverse and provides women with necessary information to use the environment securely. The study suggests limited sharing of public data and ensuring authentication following the IP addresses of the users. Restricting bot-operated avatars and abusive words to protect digital security, enabling safe modes for safe exploration for women is recommended here. Without exploring details, this study gives us a preview of how to make the Metaverse a safe space from a generalization aspect. However, it identifies the authentication issue that has been one of the biggest challenges regarding the Metaverse.

In their study, Garrido and his group [31] proposed a system where users' private data can be secured by using differential privacy protection by introducing an incognito mode in VR that significantly reduces data re-identification. It portrays phases decreasing capability with each attack which results in filtered data. It exposes a threat model combining susceptibilities of web browsers to deanonymize users using VR [5]. The potential solutions it offers are firstly Incognito mode will ensure users traceability is undetectable using local differential privacy to protect VR users. In the case of Boolean attributes, randomized response method can be applied. This Randomized “offset” values in each session are used to determine the duration through deterministic coordinate transformation. Secondly, MetaGuard features an extension that provides immediate ground truth that minimizes noise. Minimizing impact on usability and protecting VR privacy are managed by MetaGuard consisting of features like Master Toggle, Feature Toggles, privacy slide. Finally, Bounded Laplacian noise theoretically implements balance between usability and privacy. This research puts emphasis on the setup of VR and its controller instead of biological data. It cannot access the hardware and firmware level attacks in VR. Differential privacy introduces noise to the VR data that deteriorates the experience. There's substantial growth in identifying real VR users and privacy protection from the data collected from 56,082 participants.

The study conducted by Bao et al. [17] used confirmatory factor analysis (CFA) to emphasize the factors that have significant influence on the adoption of metaverse by exploring affordances of both metaverse and VR/AR and perceived risks (Privacy risk, physical risk and psychological risk). The authors focus on making the metaverse sufficiently captivating that users forget the difference between real and synthetic world. Features like scalability, ubiquity, interoperability serve as methods to stimulate realistic immersive dimension. The study contributes to grabbing the attention of investors by revealing the potential advantages and threats the

Metaverse may encounter. It initiates theoretical understanding and creates further research opportunities by exploring the factors which interest users. To evaluate the constructive measures it calculates interoperability, scalability, cross-loadings and variance inflation factor (VIF) that proves discriminant validity and the calculated VIFs shows that collinearity is not a problem. Lastly the study was conducted with limited resources and time frame which exposes the possibility of social desirability biases depending on the sample size.

The thesis titled "Man-in-the-Middle Attacks on MQTT-based IoT Networks" [30] investigates the vulnerabilities of IoT devices due to limited power and conventional security measures. The study presents a comprehensive attack strategy using a WiFi Pineapple device for Man-in-the-Middle attacks on MQTT protocol packets among IoT devices. This approach incorporates MQTT parsing, Generative Adversarial Networks (GANs), and BERT models to create and transmit malicious messages, demonstrating their effectiveness against IoT anomaly detection models. The research highlights MQTT's importance in IoT communication and explores security challenges, such as port usage and legacy issues. Additionally, the study examines IoT security mechanisms like ARTEMIS IDS, OAuth 1.0, SSL/TLS, and a layered approach. It also introduces simplified GANs for generating malicious MQTT messages and outlines future research prospects. While this is not directly related to metaverse the MITM attack model can be similar for metaverse prospect, and we need to keep this kind of attack in mind while ensuring secured authentication of metaverse universe.

The thesis titled "A survey on Metaverse: Fundamentals, security, and privacy" [25] dives into the emerging concept of metaverse. It explores the immersive virtual environment powered by extended reality, artificial intelligence, and blockchain technologies. It inspects the critical security and privacy issues inherent in the metaverse, addressing privacy invasions, technological security breaches, and scalability and interoperability challenges. The study introduces a novel metaverse architecture and assesses its key characteristics. It thoroughly examines security threats and proposes state-of-the-art solutions, emphasizing the roles of key technologies. The exploration of the metaverse in this study delves into various critical characteristics, including threats and countermeasures associated with authentication and access control. Additionally, the research scrutinizes challenges and safeguards related to data management, as well as privacy concerns and corresponding countermeasures. The metaverse's network and economy are also under examination, with a focus on identifying and mitigating potential threats. Furthermore, the thesis addresses threats to the physical world and human society within the metaverse context, proposing countermeasures for such scenarios. Governance-related threats and their countermeasures are also investigated. Lastly, the thesis concludes by outlining essential research directions aimed at advancing metaverse systems, thereby contributing to the ongoing development and security enhancement of these virtual environments. The research offers insights into the metaverse's potential and the security measures essential for its growth and integration into our digital landscape.

In the research conducted by Canbay et al. [18], the Metaverse, a digital environment fusing AI, VR, AR, and more, the privacy aspect of it is explored. In the

Metaverse, personal data powers virtual experiences. It underscores growing privacy apprehensions in the Metaverse, given the vast data collection by Metaverse Service Providers (MSPs), including biometrics and behavior patterns. In digital life, privacy is vital, as data misuse can lead to discrimination and reputation damage. The paper calls for a heightened privacy focus, proposing countermeasures like user education, regulations, and privacy-enabled tech. It cautions users to weigh the benefits against privacy risks in the data-driven Metaverse environment. While developing a secured metaverse authentication system we also need to ensure that the authentication system can ensure privacy.

The paper "Security of Virtual Reality Authentication Methods in Metaverse: An Overview" [20] focuses on the authentication methods that are used to verify the identity of users who access the metaverse using virtual reality headsets. The paper compares and analyzes four types of authentication methods: information-based, biometric, multi-model, and gaze-based. The paper evaluates these methods in terms of usability, reliability, and vulnerability. The paper also suggests some possible ways to improve the security and privacy of the authentication methods, such as using blockchain, smart contracts, and decentralized techniques. The paper concludes that multi-model authentication is the most reliable method among the ones investigated, but more studies are needed to address the challenges and limitations of this method.The paper also discusses some of the privacy and security threats that the metaverse may pose, such as data integrity, distinguishing software agents from humans, human diversity in a single world, VR headset security, social engineering attacks, doxing, espionage, stalking, and psychological attacks. The paper emphasizes the need for more research and regulation to ensure the safety and benefits of the metaverse for users.

The paper "MetaSecure: a passwordless authentication for the metaverse" the authors [34] proposes a novel authentication system for the metaverse, a set of virtual reality platforms that allow users to interact with digital assets and identities. The system, called Metasecure, aims to protect the user's online identity, assets, avatars, and accounts from various cyber threats, such as identity theft, phishing, spoofing, and harassment. Metasecure uses three layers of security: device attestation, facial recognition, and physical security keys. The system is based on the Fast Identity Online (FIDO2) specifications, which enable passwordless and cryptographic authentication. The system also provides software development kits (SDKs) for integrating the authentication mechanism into any metaverse engine or device, including VR/XR glasses. The paper demonstrates the implementation and deployment of Metasecure on a fork of VRSpace metaverse running on Babylon metaverse engine. The paper also compares Metasecure with other existing authentication methods and shows that it is faster, more secure, novel, and seamless. The paper concludes that Metasecure is the first authentication module that can implement FIDO2 based security for the metaverse and can prevent cybercrimes and protect user privacy in the digital world.

Lai et al. [33] introduced an authentication scheme for Metaverse using technology. This innovative approach addresses security concerns in the environment by implementing a multi factor cluster authentication system. The authentication process is required for the smart device in the cluster reducing network latency and server load significantly. Additionally, the system enhances security by incorporating information into the generation process and minimizing errors in single factor authentication. This data is securely stored on devices. Protected by a tamper-resistant blockchain to thwart malicious adversaries. Importantly the authors also consider scenarios such as device theft attacks, user anonymity, privacy concerns and detection of adversaries when formulating the systems robust security policies. Overall this research contributes significantly to enhancing the integrity of communications, within the evolving landscape of Metaverse while ensuring effective and secure authentication mechanisms.

In their study, Ryu et al. [24] introduced a system model leveraging blockchain technology to ensure secure communication and transparent management of user identification data within the immersive metaverse. Their novel mutual authentication scheme, blending biometric data with Elliptic Curve Cryptography (ECC), enhances security for user-server interactions and avatar-to-avatar communication. Robust informal security analyses demonstrate the scheme's resilience against threats like device theft, offline password guessing, and impersonation. Also, formal security assessments using BAN logic, ROR model, and AVISPA confirm the scheme's effectiveness in providing mutual authentication, safeguarding session keys, and preventing replay and Man-in-the-Middle (MITM) attacks. Beyond security, this scheme also reduces computation and communication costs compared to existing solutions. To conclude, Ryu et al.have made a significant stride in fortifying metaverse security, offering an efficient framework poised to enhance the user experience in these immersive virtual realms.

The authors [26] explore the vital role of blockchain in the metaverse, emphasizing its potential to transform identity management and service ecosystems. They advocate for encrypted addresses as a cornerstone for decentralized identity verification, aligning with the metaverse's decentralization ethos and reducing reliance on central authorities. They introduce the concept of dedicated spectrum allocation for metaverse and blockchain services, fostering inclusivity and innovation through 'Meta-spectrum.' Nevertheless, they confront challenges such as routing complexities without traditional IP addresses, legal considerations in encrypted networks, and the urgency of post-quantum cryptography for metaverse security. In summary, Xu et al.'s work offers an insightful roadmap for integrating blockchain into the metaverse, making notable contributions to this evolving field.

In the world of contemporary information systems, the move towards Service-Oriented Architecture (SOA) has put emphasis on service decoupling. A comprehensive investigation by Boehm et al. [4] delves into issues of implementing security in SOA, specifically concentrating on government certification and licensing through inventive web application security technologies. Drawing upon a practical case study revolving around practical case study on the specification of healthcare, this investigation emphasizes the indispensable importance of strong information technology safety especially in moving delicate medical details across regional medical systems. They assert that an internet-based architecture that constantly blends authorization onto authentication may not always meet the flexile connections mandated by new SOA systems. Various cases were analyzed by the authors in terms of the eCR security subsystem where it emerged that the architectural principles plays a significant role. The fundamental principles in this regard involve outsourcing security functions, decoupling authorization from authentication, and gradually providing security. Through this investigation, the article affords useful ideas into difficult government certification and licensing in an ever-changing SOA context.

In their paper, Fremantle and his colleagues [8] discussed security issues in the Internet of Things (IoT), a concept of their joint research [8]. The researchers address important problems which have arisen as a result of the intensive development of IoT. While concentrating primarily on issues related to protecting the privacy of users as well as their data, the paper proposes a new model of Federated Identity and Access Management (FIAM) highlighting the secure use of OAuth 2.0 in IoT. The article introduces the concept of Federated Identity and Access Management (FIAM) which uses OAuth 2.0 secure interactions within the Internet of Things (IoT) framework because it concentrates more on safeguarding user privacy and data. To carefully enable this linkage, one potential approach by the team members is thoughtful integration of OAuth 2.0 and MQTT protocol which is often preferred for IoT ecosystems. By creating a prototype using their own hands, they can test if this integration is feasible or not, as well as what could possibly be good or bad about it. This method serves as an avenue through which the researchers analyze how feasible it is to merge these two together while enlightening us with their up-coming challenges. This paper aims at addressing concerns around token renewals, scopes changes among many others regarding safety concerns associated with internet of things (IoT) gadgets so that we can improve on them in later days. To summarize, this paper highlights the need of adapting standard solutions to the peculiarities of Internet of Things world with the major stress put on user-oriented access control prevention from unauthorized disclosure.

The paper researched by Liebers and colleagues [13] talks about providing user security using Gaze based authentication. It suggests using HMDs can prove to be beneficial to refrain from bystanders with the help of built-in eye tracking features. Dynamic stimulus positioning is presented here as an example for gaze based authentication. The user can focus on the targeted object and the specification of the eye will be classified. In conclusion, the paper directs into looking for unique matrices, vestibulo-ocular movements, pupil diameter for making virtual reality more reliable. In conclusion, dynamically moving objects and interaction between them and the vergence conflict connected with the pupil diameter and imposed fatigue increases the probability of gathering unique matrices.

The researchers [11] dives into a concept where users can authenticate themselves using the 3D objects in the room and a pointer. The 3D passwords in here reduce the possibilities of shoulder surfing attacks by bystanders and improve authentication. The researchers presented an authentication randomized scheme called RoomLock that can make 3D graphical passwords. The scheme uses previous analysis and research work regarding authentication in 3D. The study consisted of three parts and the results prove that RoomLock passwords are memorable. Moreover, this scheme can tackle post-hoc attacks without randomisation and immediate attacks to some extent. Some of the limitations we have here are lack of feedback, the laser beam that was used to project into the virtual world is only observable there. It also laid down the appropriate design recommendation. In conclusion, they also talked about the possibility of RoomLock being a group authentication space. This research also provides us with options to personalize to increase memorability. It used error and trial to configure the error detection. Additionally this paper illustrates the futuristic approach 3D authentication can lead to.

The work performed by Miller and his group [23] uses the approach to make matches by pairing up between trajectory features which analyzes behavior based authentication. To prevent malicious session attacks the system uses the paired relationship to analyze every geometric discrimination. The dataset was collected by using three off-the-shelf VR systems each of them having a headset and left and right hand controllers. The users are tracked and their weight, height were recorded before asking them to indulge in tasks. 13 matches were counted from two sets of trajectories by using a training phase to learn perceptron and biasness of a value. Then the first set of data was cross examined with the second set of data to demonstrate user characteristics from right and left controller positioning and headset orientation. One limitation was that 41 users from the 46 dataset were right handed users. To eradicate this issue, front end classifiers to differentiate left and right hand users are proposed as well. In conclusion, the ball throwing is a simple behavioral activity but at the same time it is hard for the hackers to duplicate. The effectiveness of feature matches was determined by analyzing 213combination of subsets and a higher accuracy was reached by eliminating certain features furthermore.

The study carried out by Zhu and colleagues [15] presents a two factor knowledge based and biometric user authentication scheme for VR systems featured with eye tracking capabilities. It performs by allotting a secret password for users dependent on the recorded rhythms of the users blinking and is combined with a pattern of pupil size variation which is unique. Even if the blinking rhythm matches, the pupil diameter appeared distinctively in trials. BlinKey provides high security, requires no extra sensors and is hand free. For collecting data 52 people were considered with 6 training samples. The system clocks the pupil size. Two vectors represent blinking onset and offset and the interval between the time the user opens eyes and closes is timed. Relative intervals are introduced because the same user can have different blinking rhythms depending on the mood. Fast Fourier transformation is performed to extract the biometric identifier. The longer the time duration of blinkey is, the more secure it is against Zero-effort attacks, statistical attacks. One of the limitations is it requires a long time to enroll . In conclusion, Blinkey provides some error tolerance, the login time is short but it improves the privacy, usability and enhances the security and experience.

This paper [29] introduces SSI4Web, a framework designed to integrate Self-sovereign Identity (SSI) into web services, offering a passwordless authentication method that enhances user control and privacy. It leverages blockchain technology to manage digital identities securely and provides a detailed architecture with implementation details. The framework is constructed using Hyperledger Aries and Indy as basis, and within it is a threat model and requirement analysis for enhanced security. Additionally, the paper presents a use-case protocol flow of the framework, explains its compliance with the requirements thereof together with stating its benefit, limitation, and potential future work. SSI4Web is a considerable advancement for safer, user centric web services means of authentication.

Chapter 3

Proposed Model

When the user starts an authentication process by giving the user ID, the device authentication service, shown in Fig. 3.1, does some in-device operation (ensures security) and sends the request to the main server. The main server checks if the user wants to log in to this server(main server) or some other server(guest server). If the user wants to log in to another server, the main server establishes a secure authentication session with that guest server. Then it proceeds to challenge the user with a cryptographic challenge created using the user's public key which can only be solved using the private key of that user. Upon receiving the challenge user device accesses the private key through biometric authentication, uses it to create the response to the challenge, and sends it to the main server. The main server authenticates the response, if the response is correct and the user is trying to log in to this server grants the login. In case of logging into another server main server sends an authentication token and user avatar information. Finally, the guest server uses the token and avatar information to grant login to that user and completes the session.

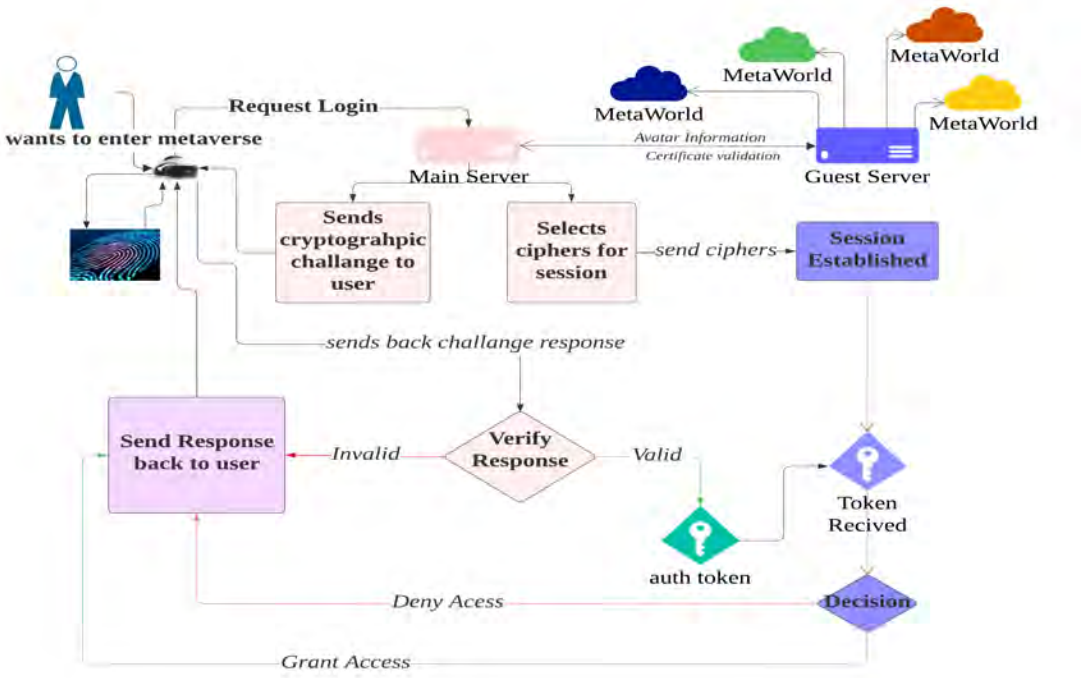


Figure 3.1: Workflow of the model

3.1 Environment

Hardware Infrastructure

The metaverse relies on a sturdy hardware infrastructure to support its operations, including high-performance servers and personal devices like VR headsets. Physical data centers house these servers, ensuring reliable connectivity and data management. Additionally, advanced networking equipment supports high-speed data transfer and low latency, enhancing the immersive experience for users.

Server Infrastructure

The server infrastructure of the metaverse consists of two main components: the main server and the guest server. The main server acts as the central hub, storing user data, managing information flow, authentication, and serving as the portal to various metaworlds. On the other hand, the guest server functions as a temporary host for users accessing metaworlds outside their usual realms, ensuring secure and streamlined navigation across the metaverse.

Metaworlds

Metaworlds are distinct, immersive virtual environments hosted on servers within the metaverse. They offer unique experiences ranging from bustling cities to serene landscapes and interactive game worlds. These metaworlds serve as digital canvases for a wide array of user activities, encouraging creativity and exploration.

User Navigation

Users navigate the metaverse by searching for metaworlds using identifiers like `username@metaworldName.com`. Direct access is provided if the desired metaworld is hosted on the main server. If the metaworld is hosted on other servers, the guest server facilitates access, ensuring users can seamlessly explore a diverse range of environments.

Authentication and Session Management

Authentication and session management in the metaverse involve a digital handshake process between the main server and the guest server. The main server transmits user information to the guest server, creating a secure channel for sharing cryptographic details. This process ensures that only authorized users can access the guest server's metaworlds. Users confirm their digital identity using biometrics or other methods, and the main server generates an encrypted authentication token, which is transferred to the guest server for validation.

Collaborative Server Operation

The collaboration between main and guest servers ensures a seamless user experience within the metaverse. Illustrated in Fig. 3.2, metaworlds hosted within the guest server serve as collaborative digital spaces for user engagement. This cooperative interaction exemplifies the fluid and interconnected nature of the digital reality within the metaverse.

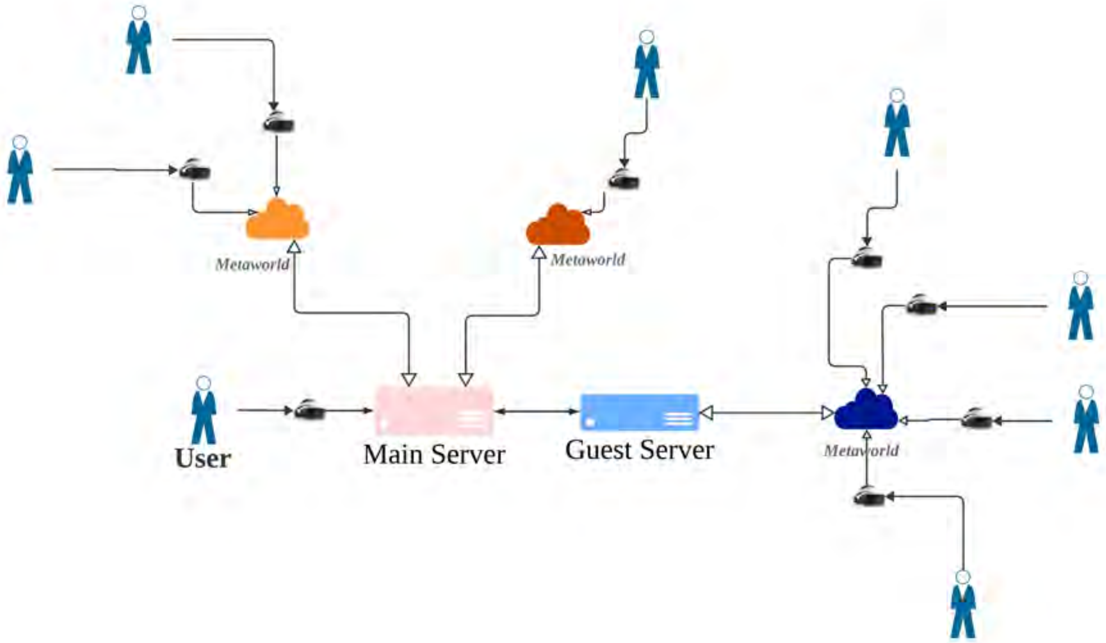


Figure 3.2: Environment

3.2 Flow of The Authentication Process

First, we define mathematical notations and data model in Table 3.1. Then, we present a flow of the authentication process to demonstrate how our proposed model will work.

Table 3.1: Cryptographic Notations I

Notation	**Description**
U	Username (Name of the user, [username@serveraddress])
M_S	Main Server
G_S	Guest Server
D	Device (User's Device)
$K_U^{M_S}$	Public key of U to be used for authentication with M_S
$K_U^{-1\|M_S}$	Private key of U to be used for authentication with M_S
A_T	Authentication Type
A_C	Authentication Challenge
$Cert_{M_S}$	Certificate of main server
$Cert_{G_S}$	Certificate of guest server

The U starts an authentication process by sending an authentication request from D to the M_S. M_S checks if the user wants to log in to M_S or G_S. If the U wants to log in to G_S, M_S establishes a secure authentication session with that G_S using the TLS encryption protocol. Then it proceeds to challenge the user with a cryptographic challenge A_C created using the public key $K_U^{(M_S)}$ which can only be solved using the private key $K_U^{-1|M_S}$ of that user. Upon receiving A_C, the user device accesses the $K_U^{-1|M_S}$ through biometric authentication, uses it to create the response to the A_T, and sends it to the M_S. M_S authenticates the response; if the response is correct and the user is trying to log in to this server, M_S grants the login. In the case of logging into G_S, the M_S gets the requirements from G_S first, then authenticates the User following that requirement. Fig. 3.3 illustrates this process. Then M_S uses a token-based approach to communicate between the user and G_S and starts a session between the U and G_S.

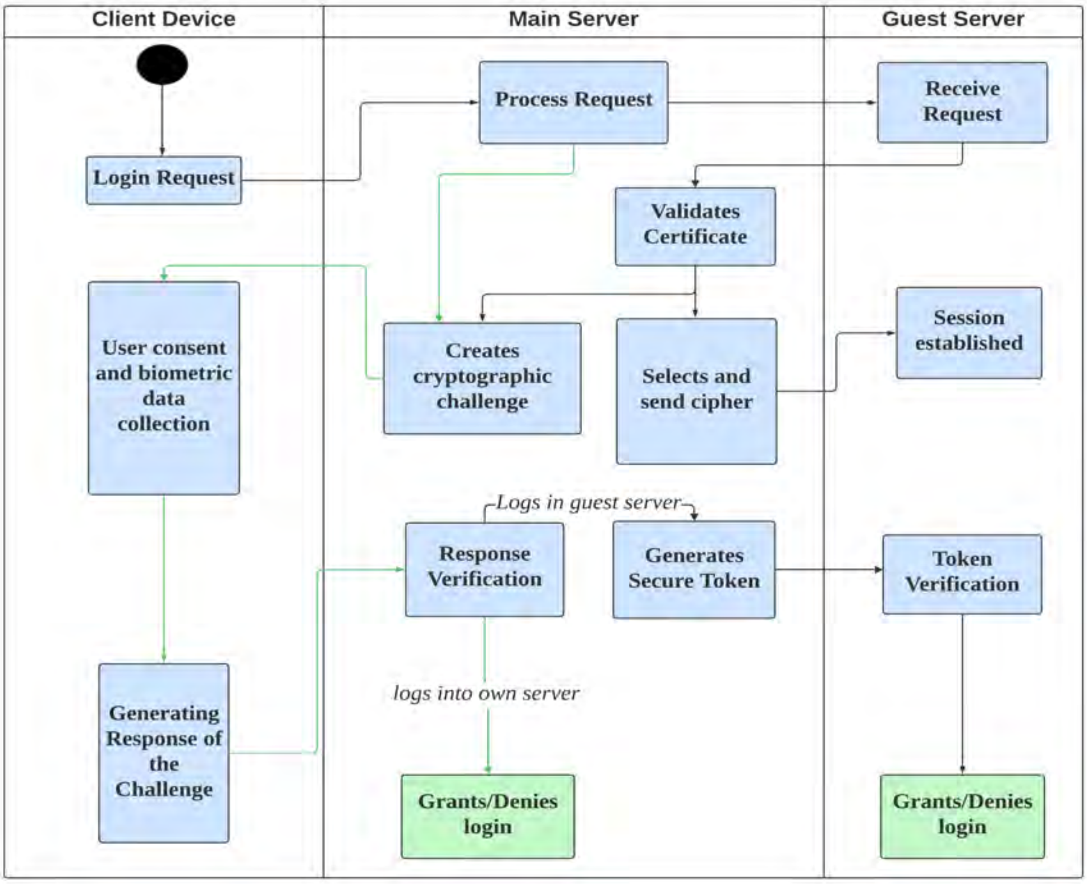


Figure 3.3: Flow of Authentication

3.2.1 Login Request

When a user initiates a login request in their D (ex. VR device), the D checks its integrity, including both hardware and system integrity checks. This ensures that the device is not compromised or altered by any external entity. After that, the device starts a temporary secure authentication mode and sends the authentication request to the M_S as illustrated in Fig. 3.4.

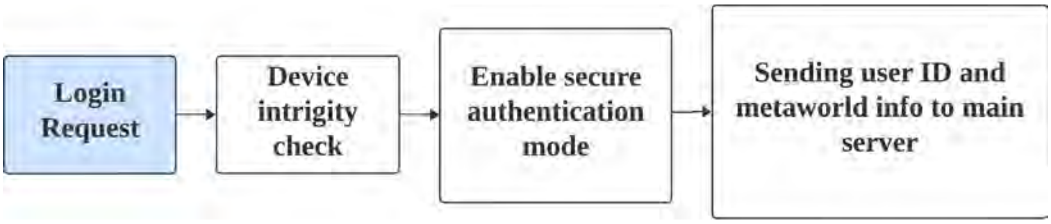

Login Request
Device intrigity check
Enable secure authentication mode
Sending user ID and metaworld info to main server

Figure 3.4: Login Request

3.2.2 Process Request

Depicted in figure 3.5, upon receiving authentication request M_S validates the request and initiates the authentication session. If the U requests to connect to this server it goes to A_C creation step. Otherwise, the server forwards the request to the G_S along with $Cert_{G_S}$.

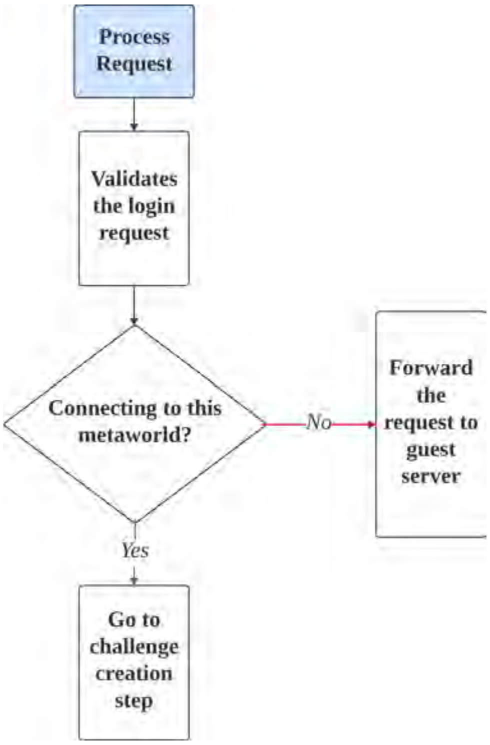

Process Request
Validates the login request
Connecting to this metaworld?
No
Forward the request to guest server
Yes
Go to challenge creation step

Figure 3.5: Process Request

3.2.3 Receive Request

The G_S receives the request sent by another server (M_S) as depictated in Fig. 3.6. It validates the $Cert_{M_S}$ so that the G_S server knows it is communicating with the right server and can make use of the public key info for further authentication communication of this session. It sends back its requirements and supported ciphers along with a $Cert_{G_S}$.

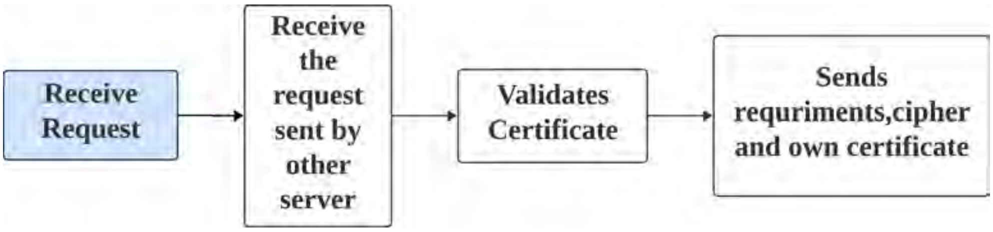

Receive Request
Receive the request sent by other server
Validates Certificate
Sends requriments,cipher and own certificate

Figure 3.6: Receive Request

3.2.4 Validates Certificate

M_S receives cipher, $Cert_{G_S}$, and requirements sent by the G_S. It validates the $Cert_{G_S}$. If the $Cert_{G_S}$ is valid, M_S does two things parallelly. It selects a cipher (offered by the G_S) and sends the response to the G_S.

3.2.5 Selection and Transmission of Cipher

M_S selects a cipher offered by the G_S and sends it to the G_S along with its certificate.

3.2.6 Session Established

When the G_S receives another response from the M_S with the selected cipher and its certificate, the G_S again checks the certificate and establishes the authentication session. The G_S waits a fixed amount of time for the response of the M_S to complete the authentication.

3.2.7 Creates a Cryptographic Challenge

he M_S uses the user ID provided by the U to find the public key $K_U^{(M_S)}$ (saved on registration) as indicated in Fig. 3.7. It uses $K_U^{(M_S)}$ to forge a cryptographic challenge A_C. For example, encrypting a random number and telling the user to decrypt this, add a certain number to it, and finally sign it. This ensures only an authentic user can solve the challenge. The M_S sends this cryptographic challenge and the certificate to the user.

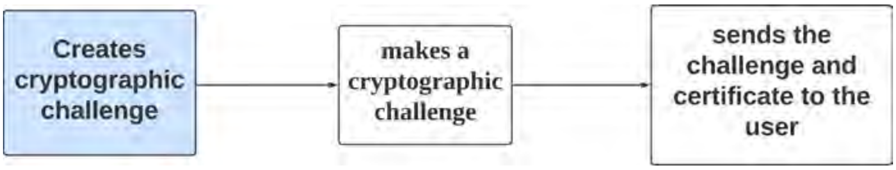

Creates cryptographic challenge
makes a cryptographic challenge
sends the challenge and certificate to the user

Figure 3.7: Creation of Cryptographic Challenge

3.2.8 Collection of Biometric Data with User Consent

In Fig. 3.8 The user device receives the A_C and validates the certificate that came along with it. The device shows a popup to confirm if the user wants to sign in to this specific meta-world. Upon user confirmation, the device scans the user's IRIS or any biometrics to confirm the identity.

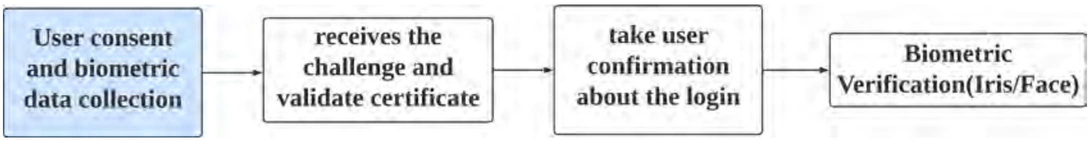

User consent and biometric data collection
receives the challenge and validate certificate
take user confirmation about the login
Biometric Verification(Iris/Face)

Figure 3.8: Biometric Data Collection

3.2.9 Generating Response to the Challenge

As demonstrated in Fig. 3.9, If the biometric identification on the device level is completed, the device accesses the $K_U^{-1|M_S}$ in a secured storage TPM. Then the device authentication service uses the $K_U^{-1|M_S}$ to decrypt and solve the challenge provided by the M_S. Finally, it signs the response and sends it back to the M_S.

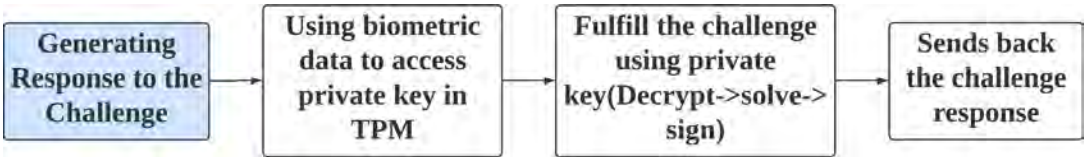

Generating Response to the Challenge
Using biometric data to access private key in TPM
Fulfill the challenge using private key(Decrypt->solve-> sign)
Sends back the challenge response

Figure 3.9: Generating Response to the Challenge

3.2.10 Verification of User Response

The M_S verifies the response of the user to challenge through sign verification and correct response. For this, M_S uses $K_U^{(M_S)}$ and stored challenge. If the response is correct, the M_S proceeds to the final step. If the user were trying to log into the M_S, it grants access completing the authentication process which is portrayed in Fig. 3.10. Otherwise, it proceeds to the token generation step.

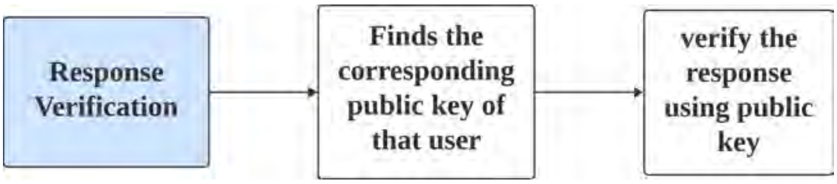

Response Verification
Finds the corresponding public key of that user
verify the response using public key

Figure 3.10: Response Verification

3.2.11 Generates Secure Auth Token

Now, in Fig. 3.11 the M_S generates an authentication token that will confirm the identity of the user to the G_S. Then the main adds the avatar information of the user and encrypts the data packet. Finally, it sends it to the G_S.

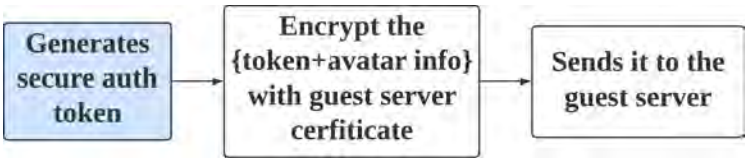

Generates secure auth token
Encrypt the {token+avatar info} with guest server cerfiticate
Sends it to the guest server

Figure 3.11: Generates Secure Auth Token

3.2.12 Token Verification

When the G_S receives an authentication token from the M_S, it decrypts and verifies the token. If the token is valid, the M_S reads the token info and the avatar info. The G_S allows the user to enter if the token confirms the identity of the user and deny otherwise which is shown in Fig. 3.12.

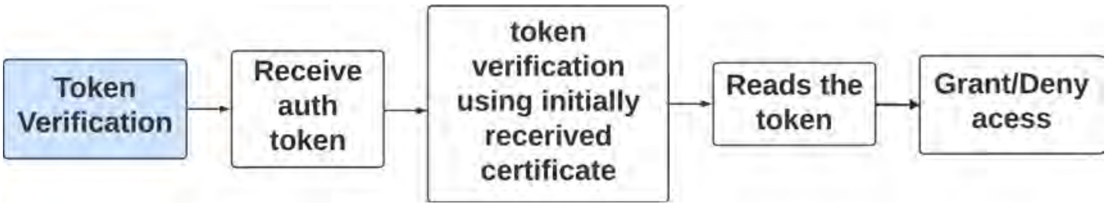


Figure 3.12: Token Verification

3.3 Session Management and Token-Based Authentication

In the realm of distributed systems, proper session management is crucial in guaranteeing the safety and effectiveness of user engagement. Our examination of the proposed system's session management will explore various key components, including token-based authentication, expiration management, centralized tracking, and a strong revocation protocol.

- **Token-Based Authentication:** In the proposed system, session management is facilitated through a token-based authentication mechanism. Upon user login, the M_S issues an authentication token encapsulating essential user information. The system promptly sends this token to the G_S which utilizes it to authenticate future requests made by this user. As a result of its short-lived nature, the token ensures secure and efficient management of user sessions.

- **Token Expiry and Renewal:** To enhance the security of the tokens, there needs to be a short expiration time. But that will require another user login which will be inconvenient for users. To address this inconvenience there needs to be a token refresh method that can allow the extension of the session's duration without requiring another user login. This method ensures continuous user engagement while also maintaining security standards.

- **Centralized Session Tracking:** The session management service's central hub is the M_S that takes care of all active sessions. M_S can work together with G_S and users in such a way they can validate tokens or fetch new session details. This guarantees that the entire system will share a streamlined session management measure.

- **Session Revocation Protocol:** The mechanism has a resilient way of cancelling any session in case it is breached or the client asks for it. The expired token status is broadcasted from the M_S to the user's device and G_S thus serving as effective session cancellation. It also preserves the security system's integrity at the same time.

3.3.1 User Authentication and Token Distribution

Upon successful authentication of a user, M_S will generate a unique access token with details concerning the person's identity and authorization. The access token is transferred securely to G_S to facilitate verification of any future requests by the user over the network. With such a token functioning as a bearer credential, the G_S does not push the user through repetitive authentication dialogues; instead, it allows them seamless entry into the system. Consequently, the conversation involving the client and G_S takes place without further authentication prompts. Therefore, more efficient communication between the G_S and users is achieved, as depicted in Fig. 3.13 as well as the token serve as a proof of user's authenticity.

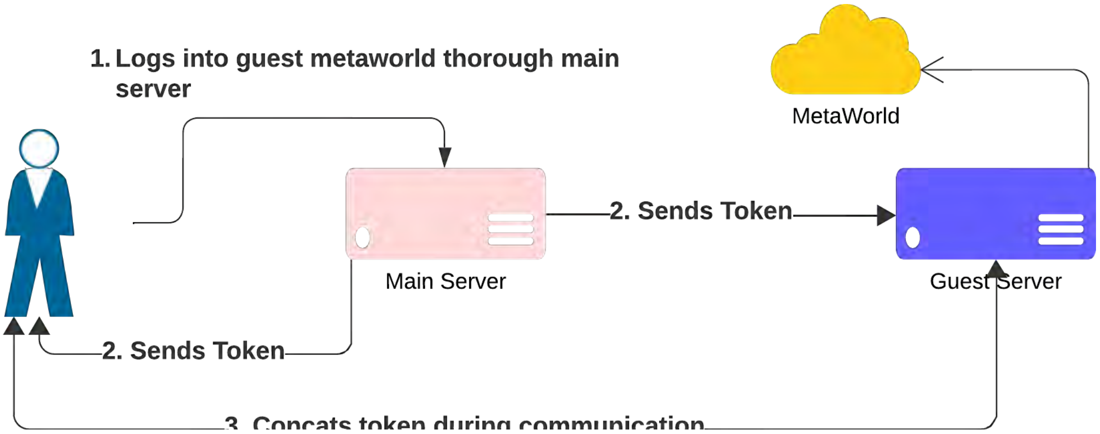

1. Logs into guest metaworld thorough main server
MetaWorld
2. Sends Token
Main Server
Guest Server
2. Sends Token
3. Concats token during communication

Figure 3.13: Token Distribution

3.3.2 Token Expiry and Broadcast of Status

A specific time limit is set for token creation to prevent any potential threat of unauthorized access through stolen tokens. If a user's session is ongoing and their token expires, the system can quickly locate an expired token and act on it as required to ensure safety. Both the user device and the G_S are informed of the token's invalidity status by M_S without delay as depicted in Fig. 3.14. This guarantees that the end of a session is communicated to every party enabling the user to do necessary activities. Such activities may comprise renewing the session through re-authentication or using the token refresh mechanism.

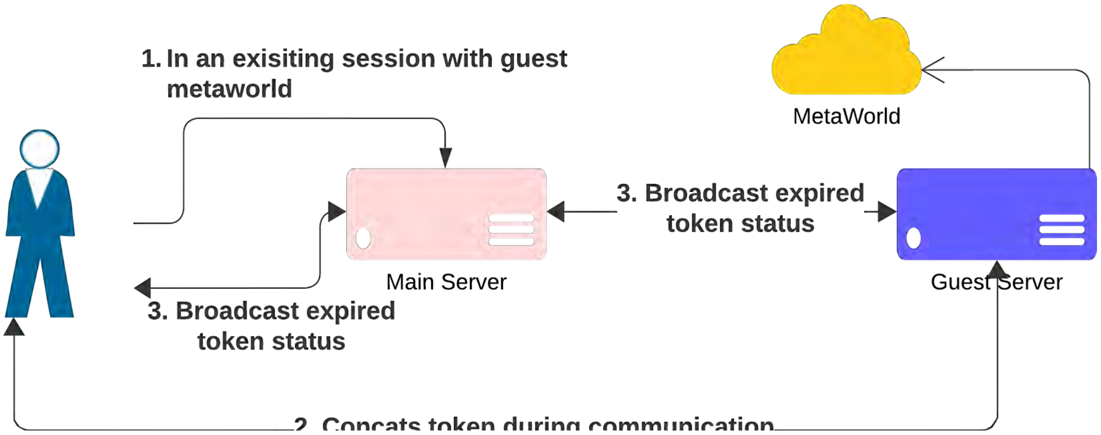

1. In an exisiting session with guest metaworld
MetaWorld
3. Broadcast expired token status
Main Server
Guest Server
3. Broadcast expired token status
2. Concats token during communication

Figure 3.14: Token Expiration

3.3.3 Session Termination and Token Invalidity Notification

A session termination process is activated the moment a user logs out or access is revoked by the M_S. While informing the G_S and the user, the M_S voids the user's token. This way, the user's session on all linked services is instantly terminated thereby rendering any more communication or transactions in the terminated session impossible. This coordinated approach, which is portrayed in Fig. 3.15 as makes sure that the user's session is safe and secure in the architecture of the system.

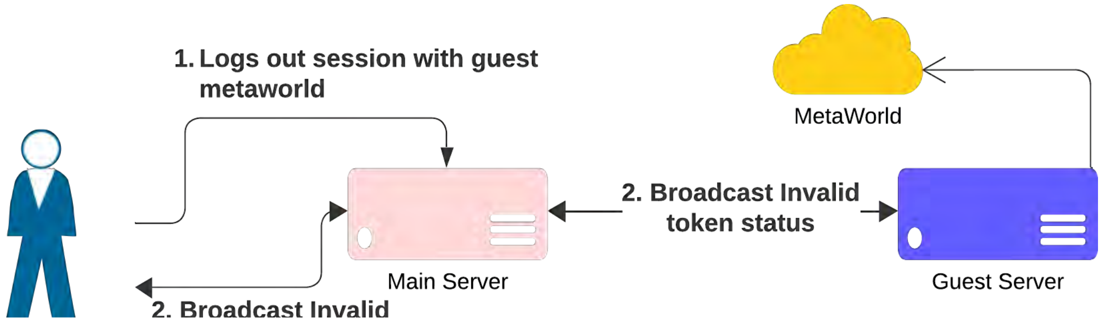

1. Logs out session with guest metaworld
MetaWorld
2. Broadcast Invalid token status
Main Server
Guest Server
2. Broadcast Invalid

Figure 3.15: Session Termination

3.4 Preliminary analysis

Authentication involves several steps, such as the user initiating communication with a server, a cryptographic challenge, and biometric authentication, among others. Each one is contributing towards strengthening the federated metaverse security in general.

- **Establishing a Secure Authentication Session:** Using cryptographic challenges during session initialization enforces security since only individuals with the right private key may authenticate successfully. This procedure makes security better by adding more complexity to it and thus reducing the chances of both brute-force and replay attempts having access any further.

- **Certificate Validation:** Bidirectional certificate validation is important in authenticating the servers that communicate. In such a process, certificates sent are checked when received. The system is then able to confirm whether the certificates have not been tampered with therefore protecting against man-in-the-middle attacks and ensuring the trustworthiness of the servers involved.

- **Authentication in Dynamic Environments:** The Metaverse changes quite fast as these virtual worlds and situations change rapidly. A major problem is how to positively identify someone within these rapid variations. The existing methods of verification are not suitable for such quick changes, and this can result in false positives or even authentication failures. Therefore, adopting an authentication system that has the ability to prompt reaction and efficiently adjust to dynamic situations is a significant challenge.

- **Cipher Selection and Secure Communication:** Choosing a cipher from the guest server's options significantly strengthens the security of communication. To ensure protection against eavesdropping and secure the privacy of important data transmitted between servers, the selection process carefully evaluates various cryptographic ciphers for data encryption. As a result, this solidifies the defense against interception and data breaches. It enables the possibility of using most up to date ciphers to ensure top of the line security of the communication.

- **Biometric Authentication:** By incorporating biometric authentication, specifically through IRIS scanning, Face Scanning, etc, an additional level of user verification is implemented that ensures decryption of the secured private keys from the TPM module happens only in the presence of the user. The biometric data never leaves the device ensuring privacy of a user.

- **Private Key Protection:** Protecting the private key is crucial in safeguarding a user's confidential data. Implementing measures such as biometric authentication and utilizing secure storage methods like Trusted Platform Module (TPM) greatly bolsters security. These precautions ensure that the private key remains safe, even in the event of a device breach, fortifying against potential attacks aiming to exploit this crucial element of data protection.

- **Challenge-Response Mechanism:** Adding a challenge-response mechanism can make the authentication process more dynamic. In addition, this can also be useful in increasing one's security because it guarantees that only the real user decrypts questions in cryptography with his or her key . Therefore, replay attacks are prevented and the user himself is involved in the process of authentication. That in turn reduces the prospect of automated or passive forms of attacks.

- **Token-based Verification:** An authentication token that contains user avatar metadata along with proper signature for that identity allows servers to set a confirmation of identity. The token comprises of secure identity attributes which are encrypted for secure transmission. Besides confirming the identity of the user, it also provides a reliable connection with the guest servers, thereby avoiding any illegitimate access by the users.

- **End-to-End Encryption:** End-to-end encryption lengths protect the data packets from interception or interference by unauthorized parties. In this way, during the process of authenticating certain information, the ability of the system to provide confidentiality and integrity of this information is guaranteed by encrypting it at the source and decrypting it only at the intended destination. This advanced safeguard will ensure that the occurrences of breaches during data transfer are greatly reduced while ensuring that there is a secure means of passing sensitive data.

- **Session Timeout:** The session timeout mechanism is convenient because there is a high risk of an attack on the user's device. Thus, by giving a limited period of time for the session with guest server, the amount of time for the system being exposed is minimized. This reduces the chances of the attacker exploiting any weaknesses any time there is an attempt at authentication hence enhancing the security of the system.

- **User Intent Confirmation:** Prior to the collection of biometric data, it is equally important to get confirmation of the user's intent to prevent abuse. It shows the action that will be happening and asks for biometric confirmation for it, which is important in case the device has been left open to the public. As such, this mode of thinking aligns authentication with the user needs and expectations to create a more secure and acceptable solution.

The initial analysis highlights the strong security capabilities of the federated metaverse authentication framework. However, in order to solidify its effectiveness and mitigate future security risks in real-world use, it is crucial to conduct thorough testing and validation, including in-depth threat modeling and vulnerability assessments. The diverse range of security measures employed work together to create a robust and reliable authentication process within the federated metaverse environment.

Chapter 4

Structure & Implementation

4.1 Data Model

Next, we will present an implementation of our proposed model to show how it could work in a practical environment. Before that we introduced mathematical notations and data model in Table 4.1 and Table 4.2 which will be used from now on.

Table 4.1: Cryptographic Notations II

Notation	**Description**
U	Username (Name of the user, [username@serveraddress])
M_S	Main Server
G_S	Guest Server
D	Device (User's Device)
D_{ID}	Device ID of User's Device
$K_U^{M_S}$	Public key of U to be used for authentication with M_S
$K_U^{-1\|M_S}$	Private key of U to be used for authentication with M_S
$\{\}_K$	Verify signature using a public key K
$\{\}_{K^{-1}}$	Signature using a private key K^{-1}
U_P	Pass username (Generated in main server after registration)
S_A	Server Address
A_T	Authentication Type
A_C	Authentication Challenge
A_{C_T}	Authentication Challenge
A_{C_D}	Authentication Challenge Details
A_A	Authentication Algorithm
S_S	Session status (Representing the status of visitor)
J_K	JWT public key
$J_{K^{-1}}$	JWT private key
J_{A_T}	JWT Access Token
J_{R_T}	JWT Refresh Token
J_{EXP}	Expiry time of JWT token
$Cert_{M_S}$	Certificate of main server
$Cert_{G_S}$	Certificate of guest server

We have a system with 3 entities M_S, G_S, D. Users VR device D that can communicate with M_S to register itself and get a U belonging to that specific M_S server. The U is structured with the following format, *username@serveraddress.* A user can be uniquely identified using this in the whole metaverse.

Table 4.2: Data Table

register $\triangleq (U, D_{ID}, K_U^{M_S}, U_P)$
access $\triangleq (U, D_{ID}, K_U^{M_S}, U_P, S_A)$
auth $\triangleq (U, U_P, D_{ID}, \{\}_{K^{-1}})$
create_user $\triangleq (S_S, D_{ID}, K_U^{M_S}, U_P)$
remote_access $\triangleq (A_T, A_A, A_C)$
generate_cryptographic_challenge $\triangleq (A_T, A_A, A_C)$
verify_cryptographic_challenge $\triangleq (A_T, A_{C_D}, \{\}_{K^{-1}}, K_U^{M_s})$
encode_token $\triangleq (U, J_T)$
access_token $\triangleq (U, J_{A_T}, J_{Exp}, \text{encode\_login\_token})$
refresh_token $\triangleq (U, J_{R_T}, J_{Exp}, \text{encode\_login\_token})$
encode_login_token $\triangleq \{\text{access\_token}, \text{refresh\_token}\}$
update_token $\triangleq (\text{access\_token})$
encode_update_token $\triangleq \{\text{access\_token}\}$
decode_access_token $\triangleq (J_{A_T})$
decode_refresh_token $\triangleq (J_{R_T})$
has_passkey $\triangleq (S_S, U, U_P)$
is_valid_device $\triangleq (S_S, U_P, D_{ID})$
has_challenge $\triangleq (S_S, U_P)$
pass_username_exists $\triangleq (S_S, U, U_P)$
visit_world $\triangleq (J_{A_T})$
get_jwt_public_key $\triangleq (J_{A_T})$

A $\boldsymbol{D}$ can send an access request to authenticate itself to a $\boldsymbol{M_S}$ and start a session with $\boldsymbol{G_S}$. After a successful session, it receives $\mathbf{J_{A_T}}$ and $\mathbf{J_{R_T}}$. It can visit the $\boldsymbol{G_S}$ directly using the ***visit_world*** function, which uses $\mathbf{J_{A_T}}$. $\mathbf{J_{A_T}}$ has a limited expiry time, which needs to be refreshed from $\boldsymbol{M_S}$ using the ***update_token*** function, taking $\mathbf{J_{R_T}}$ as input.

A $\boldsymbol{M_S}$ has mainly 5 public functions: ***create_user***, ***access***, ***auth***, ***update_token***, and ***get_jwt_public_key***, which are used in registration, remote access request to a world, authenticate the request, share an updated access token, and share JWT public key respectively. While the first 4 functions are for communication with the $\boldsymbol{D}$, the last function ***get_jwt_public_key*** is intended for $\boldsymbol{G_S}$.

$\boldsymbol{G_S}$ mainly uses the ***remote_access*** function to accept the remote connection, send requirements, and start a session. It uses the ***visit_world*** function to allow normal meta world communication while maintaining a session using the $\mathbf{J_{A_T}}$ token.

4.2 Protocol flow

Now, we will be presenting the protocol flow of our proposed theoretical system. The flow is outlined in the Fig. 4.1 and will be discussed in the following section.

i. An unregistered user cannot gain access to the system. To *register*, a user must send their U, D_{ID}, and $K_U^{M_S}$ to their M_S.

ii. The M_S will check if the user already exists in the system. If not, a new U_P will be generated for that user. Then, the user will be registered in the database, and the generated U_P will be sent back to the user, which can be used to access the system.

iii. Upon an access request, the M_S will check the validity of that request. If successful, the M_S will then check the S_A of that request. If the S_A contains the address of a remote G_S, it will send the U to that G_S. Otherwise, it will continue to the process of *generate_cryptographic_challenge* to complete the process locally.

iv. From the Fig. 4.2 we can see that the G_S will check the $Cert_{(M_S)}$, and if it's valid, it will check whether the received U is blacklisted. If the U is not blacklisted, it will check the S_s to identify if the user is already in session. If not, it will set the $S_s = pending$ and send the cryptographic requirements, which include (A_T, A_A, C_T), to the M_S.

v. When the M_S receives the requirements from the G_S, it will check the validity of the $Cert_{(G_S)}$, and if that's valid, continue to *generate_cryptographic_challenge*.

vi. The *generate_cryptographic_challenge* will be sent to the user, where they will sign this challenge using $K_U^{-1|M_S}$ and send it back to the M_S.

vii. The M_S will verify this signed challenge using the public key $K_U^{M_S}$. If verification is successful, it will send a session acceptance request to the G_S if the S_A contained the address of a guest server; otherwise, it will start the session locally and generate JWT Tokens.

viii. After receiving the session acceptance request, the G_S will start the session and request the J_K from M_S.

ix. The G_S admin will need to change $S_S = accepted$ to allow access to the remote user.

x. The M_S will provide the J_K to the G_S. G_S will store this in their database and send a successful response.

xi. The M_S will now generate JWT Tokens which consists of J_{A_T} and J_{R_T} as displayed in Fig. 4.3. It is signed using $J_{K^{-1}}$. The resulting JWT Tokens will be sent to the U.

xii. The U will store the JWT Tokens. Now, the received J_{A_T} can be used to access the desired server. Based on the S_A on this token, it will go to the desired server, and upon the successful validation of this token, the user will be granted access.

xiii. There is a time limit for the validity of the J_{A_T}. To renew the validity, the user can send the J_{R_T} to their M_S, which will generate a new J_{A_T} and send it back to the user.

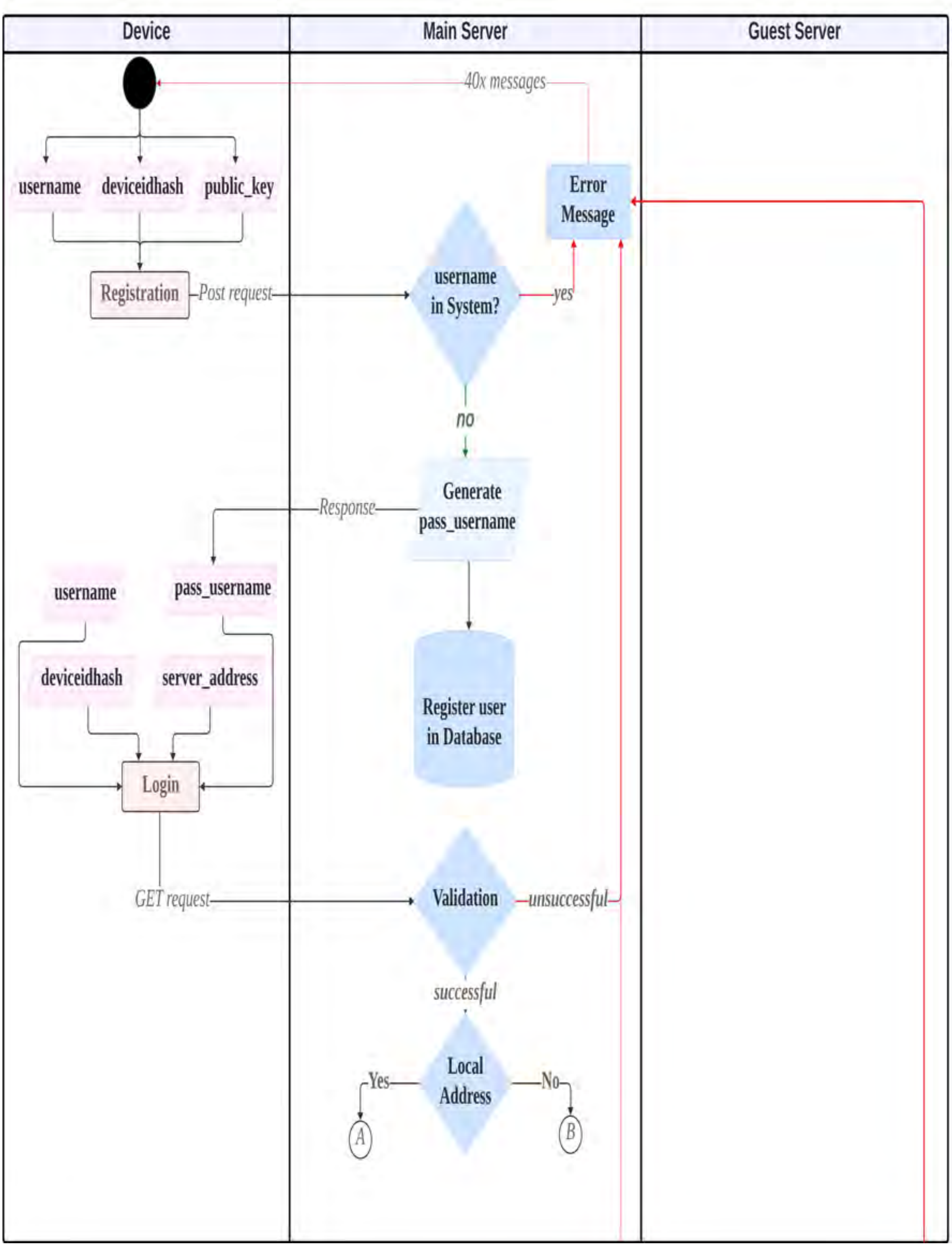


Figure 4.1: Flow Diagram

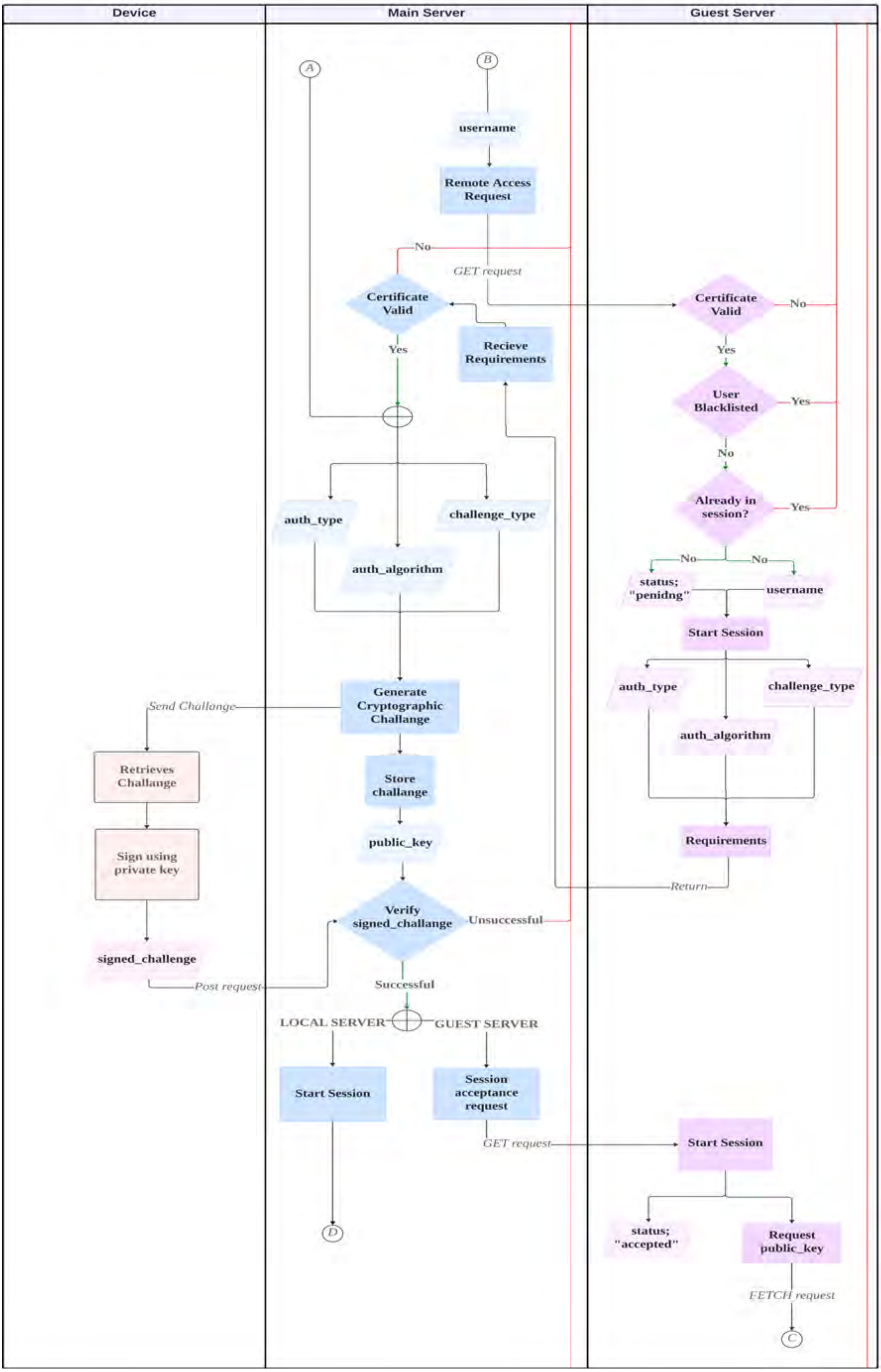


Figure 4.2: Flow Diagram continued

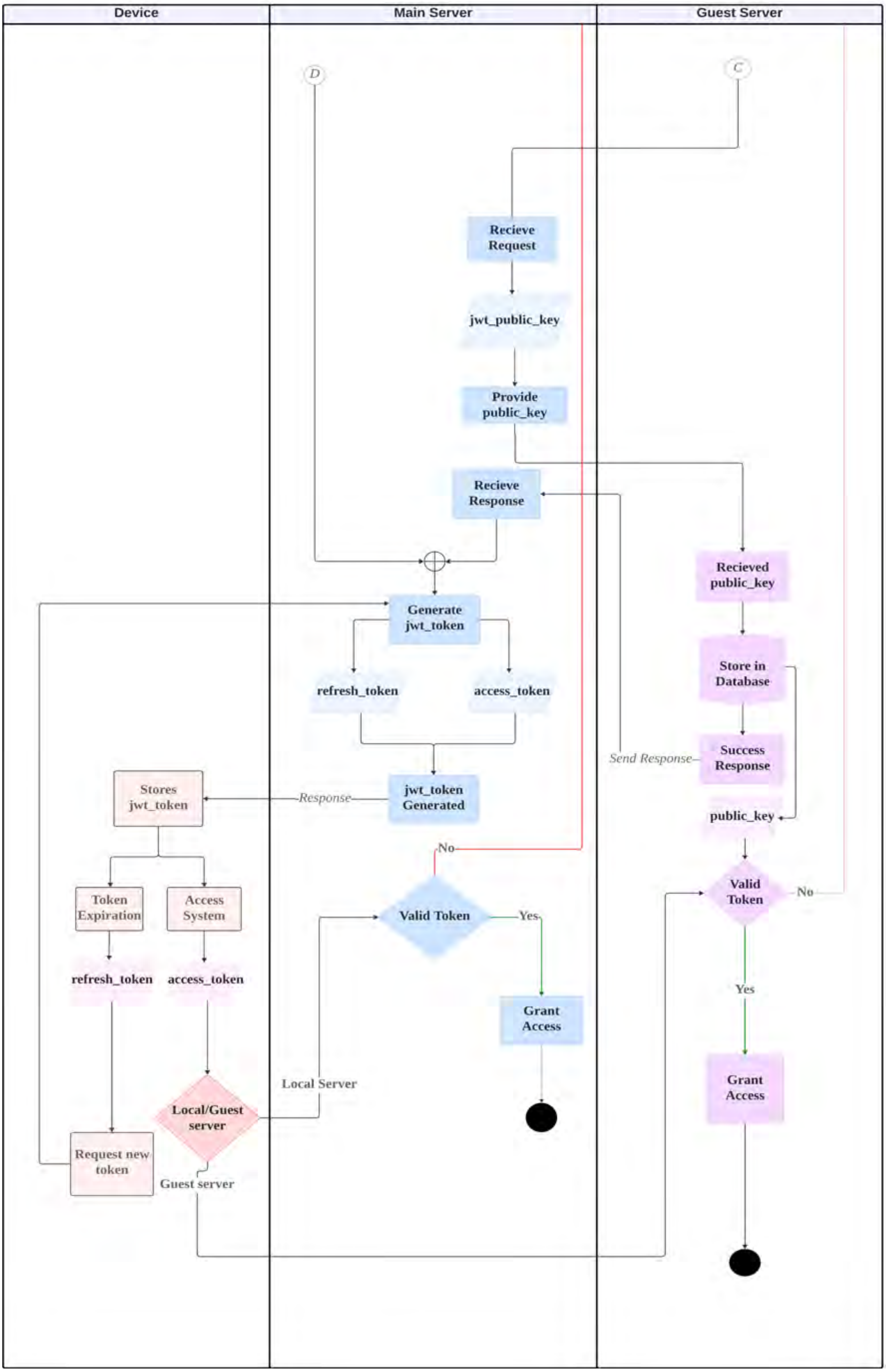


Figure 4.3: Flow Diagram continued

4.3 Algorithms

In this part of the work, we describe algorithms used for cryptographic challenges, responses of that cryptographic challenge, JWT token creation and validation.

4.3.1 Cryptographic Challenge

The algorithm(1) CreateChallenge is responsible for generating a cryptographic challenge, which includes a random number and challenge details. The challenge is then sent to the user.

Algorithm 1 CreateChallenge

```
Input: $A_T$, $A_A$, $A_{C_T}$
Output: $A_C$ (Challenge: random number), $A_{C_D}$

$A_C \leftarrow$ generate_cryptographic_challenge() $\leftarrow$ GenRandNum(fixedLength)
$A_{C_D} \leftarrow$ GetChallengeDetails()
SendChallengeToUser($A_C$, $A_{C_D}$)
return $A_C$, $A_{C_D}$
```

4.3.2 Cryptographic Response

The algorithm(2) VerifySignature is used to verify the signature of the cryptographic response by decrypting the challenge using the provided public and private keys. The algorithm(3) VerifyResponse utilizes VerifySignature to validate the cryptographic challenge response.

Algorithm 2 VerifySignature

```
Input: $A_C$, $\{A_C\}_{K^{-1}}$, $\{\}_K$
Output: isValid (a boolean indicating whether the signature is valid)

decryptedChallenge $\leftarrow$ DecryptWithPublicKey($\{A_C\}_{K^{-1}}$, $\{\}_K$)
if decryptedChallenge = $A_C$ then
    isValid $\leftarrow$ True
else
    isValid $\leftarrow$ False
end if
return isValid
```

Algorithm 3 VerifyResponse

```
Input: $A_C$, $\{A_C\}_{K^{-1}}$, $\{\}_K$
Output: isValid (a boolean indicating whether the verification was successful)

isValid $\leftarrow$ verify_cryptographic_challenge() $\leftarrow$ VerifySignature($A_C$, $\{A_C\}_{K^{-1}}$, $\{\}_K$)
return isValid
```

4.3.3 JWT Token Creation

The algorithm(4) ENCODE_LOGIN_TOKEN creates an access token and a refresh token for a user. The algorithm(5) ENCODE_UPDATE_TOKEN generates a new access token from a refresh token. The algorithm(6) ENCODE_TOKEN is used to create a JWT token by encoding the payload and signing it.

Algorithm 4 encode_login_token

```
Input: $U$, $J_{K^{-1}}$
Output: $J_{A_T}$, $J_{R_T}$

$J_{A_T} \leftarrow$ encode_token({$U$, access_token, TIME()+1min}, $J_{K^{-1}}$)
$J_{R_T} \leftarrow$ encode_token({$U$, refresh_token, TIME()+720min}, $J_{K^{-1}}$)
return $J_{A_T}$, $J_{R_T}$
```

Algorithm 5 encode_update_token

```
Input: $J_{R_T}$
Output: $J_{A_T}$

$U \leftarrow$ GetUserName() $\leftarrow$ decode_refresh_token($J_{R_T}$)
$J_{A_T} \leftarrow$ encode_token({$U$, access_token, TIME()+1min}, $J_{K^{-1}}$)
return $J_{A_T}$
```

Algorithm 6 encode_token

```
Input: payload (data to be included in the token), $J_{K^{-1}}$
Output: token (the generated JWT token)

header ← GenerateHeader()
encodedPayload ← Base64Encode(payload)
signature ← Sign(header + ”.” + encodedPayload, $J_{K^{-1}}$)
token ← header + ”.” + encodedPayload + ”.” + signature
return token
```

4.3.4 JWT Token Validation

The algorithm(7) DECODE_ACCESS_TOKEN decodes and validates an access token. The algorithm(8) DECODE_REFRESH_TOKEN decodes and validates a refresh token. The algorithm(9) ValidateJWTToken checks the validity of a JWT token by verifying its structure and signature.

Algorithm 7 decode_access_token

```
Input: $J_{A_T}$, $J_K$
Output: $U$, $J_{A_T}$, $J_{Exp}$

if ValidateJWTToken then
    $U$, $J_{A_T}$, $J_{Exp} \leftarrow$ Base64Decode(payload)
else
    Raise:InvalidSignature
end if
return $U$, $J_{A_T}$, $J_{Exp}$
```

Algorithm 8 decode_refresh_token

```
Input: $J_{R_T}$, $J_K$
Output: $U$, $J_{R_T}$, $J_{Exp}$

if ValidateJWTToken then
    $U$, $J_{R_T}$, $J_{Exp} \leftarrow$ Base64Decode(payload)
else
    Raise:InvalidSignature
end if
return $U$, $J_{R_T}$, $J_{Exp}$
```

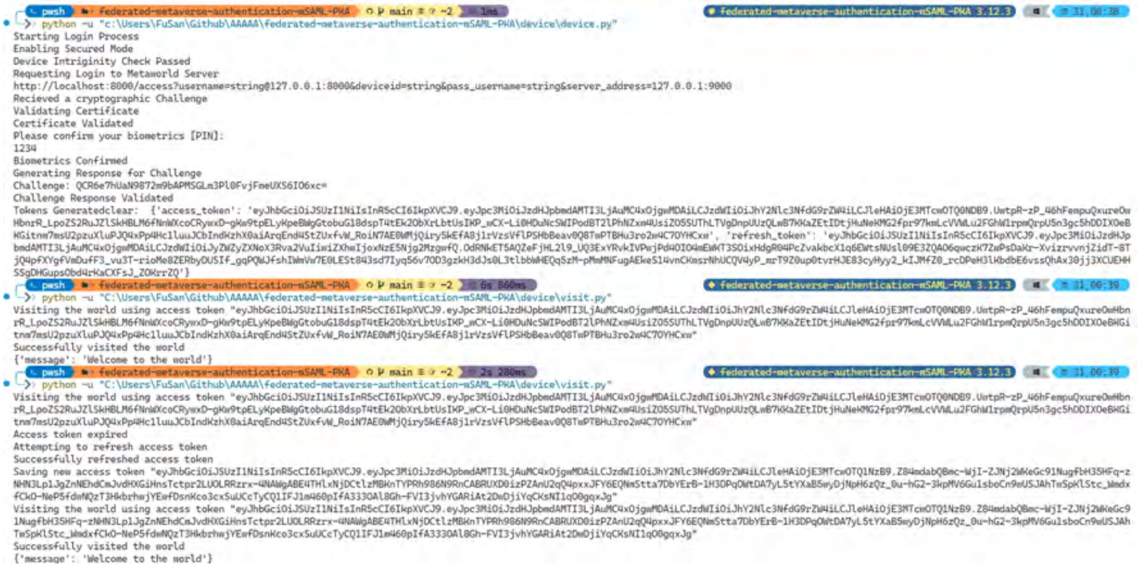
Algorithm 9 ValidateJWTToken

```
Input: token (the JWT token), secret (key for token decryption)
Output: isValid (a boolean indicating whether the token is valid)

parts ← Split(token, ”.”)
if parts has length 3 then
    header ← parts[0]
    payload ← parts[1]
    signature ← parts[2]
    expectedSignature ← Sign(header + ”.” + payload, secret)
    if signature equals expectedSignature then
        isValid ← True
    else
        isValid ← False
    end if
else
    isValid ← False
end if
return isValid
```

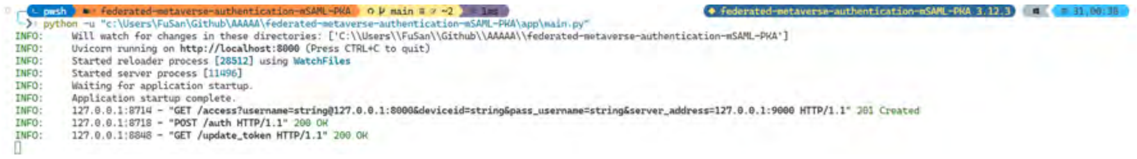
4.4 Demonstration

Our proposed version of the metaworld contains of three entites which are the D, the M_S and the G_S. In our implementation, the M_S and the G_S are built using the Python FAST API, which uses different URL endpoints to take different requests. Additionally, the M_S & the G_S have distinct databases. The database for the M_S comprise of two different tables which are called the Users table & the Challenges table as portrayed in Fig. 4.6. Similarly, the database of the G_S also includes two different tables, the Visitors table and the Blacklist table as shown in Fig. 4.8. Furthermore, public-private keypair for JWT token are stored in the M_S. The D stores J_{A_T}, J_{R_T} and $K_U^{-1|M_S}$ for user authentication. Subsequently, by focusing on the Fig. 4.4, we observe that a login request was inititated from the D to the M_S. After enabling secure mode & the passing of device's integrity check, the request is received by the M_S, which then generates a cryptographic challenge using the $K_U^{M_S}$. The user completes the response to this challenge using their PIN. After the response is verified at the M_S, a JWT token is generated which includes J_{A_T} & J_{R_T}. Following that, by using the J_{A_T} the user visits the guest metaworld. However, after some time when the user again tries to visit the guest metaworld, we notice that the J_{A_T} has expired. Then the D attempts to refresh the J_{A_T} using the J_{R_T}. Afterwards, the J_{A_T} is refreshed in the M_S and then the new J_{A_T} is saved at the D. It now allows the user to successfully visit the guest metaworld again using their renewed J_{A_T} as illustrated at Fig. 4.7.



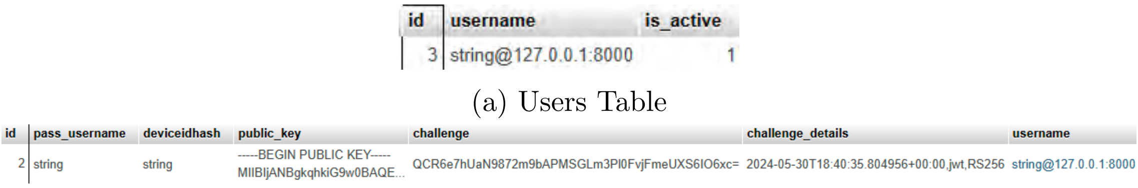
```
 pwsh  federated-metaverse-authentication-mSAML-PKA  main ≡ ? ~2  1ms        federated-metaverse-authentication-mSAML-PKA 3.12.3    31,00:38
 > python -u "c:\Users\FuSan\Github\AAAAA\federated-metaverse-authentication-mSAML-PKA\device\device.py"
Starting Login Process
Enabling Secured Mode
Device Intriginity Check Passed
Requesting Login to Metaworld Server
http://localhost:8000/access?username=string@127.0.0.1:8000&deviceid=string&pass_username=string&server_address=127.0.0.1:9000
Recieved a cryptographic Challenge
Validating Certificate
Certificate Validated
Please confirm your biometrics [PIN]:
1234
Biometrics Confirmed
Generating Response for Challenge
Challenge: QCR6e7hUaN9872m9bAPMSGLm3Pl0FvjFmeUXS6IO6xc=
Challenge Response Validated
Tokens Generatedclear:  {'access_token': 'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJpbmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJhY2Nlc3NfdG9rZW4iLCJleHAiOjE3MTcwOTQ0NDB9.UwtpR-zP_46hFempuQxureOw
HbnrR_LpoZS2RuJZlSkHBLM6fNnWXcoCRywxD-gKw9tpELyKpeBWgGtobuGl8dspT4tEk2ObXrLbtUsIKP_wCX-Li0HDuNcSWIPodBT2lPhNZxm4UsiZO5SUThLTVgDnpUUzQLwB7KKaZEtIDtjHuNeKMG2fpr97kmLcVVWLu2FGhW1rpmQrpU5n3gc5hDDIXOeB
KGitnm7msU2pzuXluPJQ4xPp4Hc1luuJCbIndKzhX0aiArqEnd4StZUxfvW_RoiN7AE0WMjQiry5kEfA8j1rVzsVflPSHbBeav0Q8TwPTBHu3ro2w4C7OYHCxw', 'refresh_token': 'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJp
bmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJyZWZyZXNoX3Rva2VuIiwiZXhwIjoxNzE5Njg2MzgwfQ.OdRNkET5AQZeFjHL2l9_UQ3ExYRvkIVPwjPd4OIO4mEWKT3SDixHdgR04PcZvakbcX1q6EWtsNUsl09E3ZQAO6qwczK7ZwPsDaKr-XvizrvvnjZidT-8T
jQ4pfXYgfVmDufF3_vu3T-rioMe8ZERbyOUSIf_gqPQWJfshIWmVw7E0LESt843sd7Iyq56v7OD3gzkH3dJs0L3tlbbWHEQq5zM-pMmMNFugAEkeS14vnCKmsrNhUCQV4yP_mrT9Z0up0tvrHJE83cyHyy2_kIJMfZ0_rcDPeH3lKbdbE6vssQhAx30jj3XCUEHH
SSgDHGupsObd4rKaCXFsJ_ZOKrrZQ'}
 pwsh  federated-metaverse-authentication-mSAML-PKA  main ≡ ? ~2  6s 860ms        federated-metaverse-authentication-mSAML-PKA 3.12.3    31,00:39
 > python -u "C:\Users\FuSan\Github\AAAAA\federated-metaverse-authentication-mSAML-PKA\device\visit.py"
Visiting the world using access token "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJpbmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJhY2Nlc3NfdG9rZW4iLCJleHAiOjE3MTcwOTQ0NDB9.UwtpR-zP_46hFempuQxureOwHbn
rR_LpoZS2RuJZlSkHBLM6fNnWXcoCRywxD-gKw9tpELyKpeBWgGtobuGl8dspT4tEk2ObXrLbtUsIKP_wCX-Li0HDuNcSWIPodBT2lPhNZxm4UsiZO5SUThLTVgDnpUUzQLwB7KKaZEtIDtjHuNeKMG2fpr97kmLcVVWLu2FGhW1rpmQrpU5n3gc5hDDIXOeBKGi
tnm7msU2pzuXluPJQ4xPp4Hc1luuJCbIndKzhX0aiArqEnd4StZUxfvW_RoiN7AE0WMjQiry5kEfA8j1rVzsVflPSHbBeav0Q8TwPTBHu3ro2w4C7OYHCxw"
Successfully visited the world
{'message': 'Welcome to the world'}
 pwsh  federated-metaverse-authentication-mSAML-PKA  main ≡ ? ~2  2s 280ms        federated-metaverse-authentication-mSAML-PKA 3.12.3    31,00:39
 > python -u "C:\Users\FuSan\Github\AAAAA\federated-metaverse-authentication-mSAML-PKA\device\visit.py"
Visiting the world using access token "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJpbmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJhY2Nlc3NfdG9rZW4iLCJleHAiOjE3MTcwOTQ0NDB9.UwtpR-zP_46hFempuQxureOwHbn
rR_LpoZS2RuJZlSkHBLM6fNnWXcoCRywxD-gKw9tpELyKpeBWgGtobuGl8dspT4tEk2ObXrLbtUsIKP_wCX-Li0HDuNcSWIPodBT2lPhNZxm4UsiZO5SUThLTVgDnpUUzQLwB7KKaZEtIDtjHuNeKMG2fpr97kmLcVVWLu2FGhW1rpmQrpU5n3gc5hDDIXOeBKGi
tnm7msU2pzuXluPJQ4xPp4Hc1luuJCbIndKzhX0aiArqEnd4StZUxfvW_RoiN7AE0WMjQiry5kEfA8j1rVzsVflPSHbBeav0Q8TwPTBHu3ro2w4C7OYHCxw"
Access token expired
Attempting to refresh access token
Successfully refreshed access token
Saving new access token "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJpbmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJhY2Nlc3NfdG9rZW4iLCJleHAiOjE3MTcwOTQ1NzB9.Z84mdabQBmc-WjI-ZJNj2WKeGc91NugfbH35HFq-z
NHN3Lp1JgZnNEhdCmJvdHXGiHnsTctpr2LUOLRRzrx-4NAWgABE4THlxNjDCtlzMBKnTYPRh986N9RnCABRUXD0izPZAnU2qQ4pxxJFY6EQNmStta7DbYErB-1H3DPqOWtDA7yL5tYXaB5wyDjNpH6zQz_0u-hG2-3kpMV6GulsboCn9wUSJAhTwSpKlStc_Wmdx
fCkO-NeP5fdwNQzT3HkbrhwjYEwfDsnKco3cxSuUCcTyCQ1IFJ1m460pIfA333OAl8Gh-FVI3jvhYGARiAt2DwDjiYqCKsNI1qO0gqxJg"
Visiting the world using access token "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdHJpbmdAMTI3LjAuMC4xOjgwMDAiLCJzdWIiOiJhY2Nlc3NfdG9rZW4iLCJleHAiOjE3MTcwOTQ1NzB9.Z84mdabQBmc-WjI-ZJNj2WKeGc9
1NugfbH35HFq-zNHN3Lp1JgZnNEhdCmJvdHXGiHnsTctpr2LUOLRRzrx-4NAWgABE4THlxNjDCtlzMBKnTYPRh986N9RnCABRUXD0izPZAnU2qQ4pxxJFY6EQNmStta7DbYErB-1H3DPqOWtDA7yL5tYXaB5wyDjNpH6zQz_0u-hG2-3kpMV6GulsboCn9wUSJAh
TwSpKlStc_WmdxfCkO-NeP5fdwNQzT3HkbrhwjYEwfDsnKco3cxSuUCcTyCQ1IFJ1m460pIfA333OAl8Gh-FVI3jvhYGARiAt2DwDjiYqCKsNI1qO0gqxJg"
Successfully visited the world
{'message': 'Welcome to the world'}
```

Figure 4.4: Device Demonstration



```
 pwsh  federated-metaverse-authentication-mSAML-PKA  main ≡ ? ~2  1ms        federated-metaverse-authentication-mSAML-PKA 3.12.3    31,00:38
 > python -u "c:\Users\FuSan\Github\AAAAA\federated-metaverse-authentication-mSAML-PKA\app\main.py"
INFO:     Will watch for changes in these directories: ['C:\\Users\\FuSan\\Github\\AAAAA\\federated-metaverse-authentication-mSAML-PKA']
INFO:     Uvicorn running on http://localhost:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [28512] using WatchFiles
INFO:     Started server process [11496]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:8714 - "GET /access?username=string@127.0.0.1:8000&deviceid=string&pass_username=string&server_address=127.0.0.1:9000 HTTP/1.1" 201 Created
INFO:     127.0.0.1:8718 - "POST /auth HTTP/1.1" 200 OK
INFO:     127.0.0.1:8848 - "GET /update_token HTTP/1.1" 200 OK
```

Figure 4.5: Main Server Demonstration



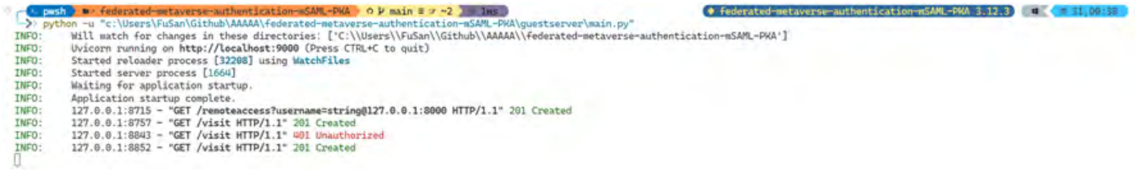
id	username	is_active
3	string@127.0.0.1:8000	1

(a) Users Table

id	pass_username	deviceidhash	public_key	challenge	challenge_details	username
2	string	string	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQE...	QCR6e7hUaN9872m9bAPMSGLm3Pl0FvjFmeUXS6IO6xc=	2024-05-30T18:40:35.804956+00:00,jwt,RS256	string@127.0.0.1:8000

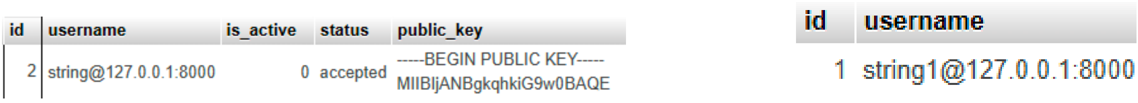
(b) Challenges

Figure 4.6: Main Server Database



```
 pwsh  federated-metaverse-authentication-mSAML-PKA  main ≡ ~2  1ms                    federated-metaverse-authentication-mSAML-PKA 3.12.3   31,00:38
> python -u "c:\Users\FuSan\Github\AAAAA\federated-metaverse-authentication-mSAML-PKA\guestserver\main.py"
INFO:     Will watch for changes in these directories: ['C:\\Users\\FuSan\\Github\\AAAAA\\federated-metaverse-authentication-mSAML-PKA']
INFO:     Uvicorn running on http://localhost:9000 (Press CTRL+C to quit)
INFO:     Started reloader process [32208] using WatchFiles
INFO:     Started server process [1664]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:8715 - "GET /remoteaccess?username=string@127.0.0.1:8000 HTTP/1.1" 201 Created
INFO:     127.0.0.1:8757 - "GET /visit HTTP/1.1" 201 Created
INFO:     127.0.0.1:8843 - "GET /visit HTTP/1.1" 401 Unauthorized
INFO:     127.0.0.1:8852 - "GET /visit HTTP/1.1" 201 Created
```

Figure 4.7: Guest Server Demonstration



id	username	is_active	status	public_key
2	string@127.0.0.1:8000	0	accepted	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQE

(a) Visitors Table

id	username
1	string1@127.0.0.1:8000

(b) Blacklist

Figure 4.8: Guest Server Database

Chapter 5

Security Analysis

5.1 CIA Analysis

The “CIA triad” stands for Confidentiality, Integrity, and Availability. It is a ubiquitous model that forms the basis for developing security systems. Confidentiality, Integrity, and availability of data in a system are crucial. The CIA analysis is a standard practice used to find vulnerabilities in a security system and find potential solutions to them.

5.1.1 Confidentiality

Confidentiality refers to the availability of data to the authorized person only. When the system transfers data, it must use some mechanism so that even if an unauthorized person gets access to the data they can’t retrieve the information from it, in other words, the data remains confidential. To ensure confidentiality, the proposed authentication model uses Transport Layer Security(TLS) for all data communication. TLS is a standard practice for building secure web systems using state-of-the-art encryption(a combination of both symmetric and asymmetric encryption algorithms) systems. This protocol encrypts the communication between sender and receiver maintaining the confidentiality of data. Furthermore, TLS ensures the authenticity and Integrity of data. It makes use of a TLS certificate provided by the CA(certificate authority) to authenticate the user/server and MAC(Message Authentication Code) to ensure the integrity of data. Confidentiality is ensured by:

- **Encryption with TLS:** Use TLS to encrypt all communication between the user, main server, and guest server. TLS provides encryption through a combination of symmetric and asymmetric encryption algorithms. Symmetric encryption is used for bulk data encryption, while asymmetric encryption is used for key exchange and authentication. This encryption ensures that even if an unauthorized person gains access to the data, they cannot retrieve the information from it, maintaining confidentiality [10]. In TLS (Transport Layer Security), public key cryptography is primarily used for key exchange, authentication, and ensuring data integrity.

    - **RSA (Rivest-Shamir-Adleman):** RSA is widely used in TLS for key exchange and digital signatures. During the TLS handshake, RSA key exchange allows the client and server to establish a shared secret for symmetric encryption. RSA digital signatures are used for server authentication and to ensure the integrity of the handshake messages.

  - **Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH):** Diffie-Hellman and its elliptic curve variant, ECDH, are used for key exchange in TLS. These algorithms allow the client and server to securely negotiate a shared secret over an insecure channel without exchanging the secret itself.
  - **Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA):** DSA and ECDSA are used for digital signatures in TLS. They provide a way for entities to verify the authenticity and integrity of messages exchanged during the TLS handshake.

- **Authentication with TLS Certificates:** Use TLS certificates provided by a Certificate Authority (CA) to authenticate the user and servers. TLS certificates ensure the authenticity of the communicating parties, preventing man-in-the-middle attacks and unauthorized access. Proper validation of TLS certificates is essential to maintain confidentiality and security [5].

- **Message Authentication Code (MAC):** Use MAC to ensure the integrity of data transmitted over the TLS-encrypted connection. MAC verifies that the data has not been altered during transmission. It adds an extra layer of security to maintain data integrity and confidentiality [2].

- **Perfect Forward Secrecy (PFS):** Implement Perfect Forward Secrecy (PFS) to ensure that each session key is unique and not derived from any long-term secret. PFS enhances confidentiality by ensuring that even if a long-term secret key is compromised, past sessions remain secure [1].

5.1.2 Integrity

Integrity of data means that the data is not altered in any way, it's trustworthy and free from tempering. It's a crucial part of the CIA triad. If proper integrity checking of data is not in place in an authentication system, the system is vulnerable to active Man in the Middle Attack. Our authentication system has multiple ways to maintain the integrity of the data in the entire process of authentication. Firstly, there's a device level integrity check by the operating system of that device to ensure it's not compromised. Then the system uses TLS encryption for all data exchange which ensures data integrity using MAC. But for sensitive data like cryptographic challenge and response the system makes use of public key cryptography. It uses digital signature to ensure data is coming from a valid source and not altered along the way. This also helps prevent replay attacks. Integrity is ensured by:

- **Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA):** DSA and ECDSA are used for digital signatures in TLS to verify the authenticity and integrity of messages exchanged during the TLS handshake [10].

- **Public Key Cryptography:** For sensitive data like cryptographic challenges and responses, TLS employs public key cryptography to ensure data integrity. Public key cryptography uses digital signatures to verify that the data is coming from a valid source and has not been altered along the way, preventing tampering and maintaining integrity [1].

5.1.3 Availability

The last but not least component of the CIA triad is availability. Even if the system's confidentiality and integrity are maintained it must be available to the users to access it and use it. Otherwise, building a very secure system is meaningless or to some extent useless. Our proposed model is federated in nature where the metaverse consists of a lot of federated meta worlds and each of them can act independently. This decentralized nature of the metaverse in our model protects the metaverse from large-scale DDoS attacks. This ensures that even if some meta-worlds face DDoS attacks despite having a DDoS protection system of their own, other meta-worlds will be available for the users. Availability is ensured by:

- **Redundancy and Load Balancing:** Implement redundancy and load balancing mechanisms within the TLS infrastructure to distribute authentication requests across multiple servers. This ensures that if one server fails or becomes overloaded, other servers can handle the traffic, maintaining availability [6].
- **Distributed Denial of Service (DDoS) Protection:** Deploy DDoS protection mechanisms within the TLS infrastructure to detect and mitigate DDoS attacks aimed at the authentication system. This includes rate limiting, traffic filtering, and the use of specialized DDoS protection services [12].
- **Fault Tolerance:** Design the TLS infrastructure with fault tolerance in mind, ensuring that it can continue to operate even in the event of hardware failures or network disruptions. Implement failover mechanisms and redundant data storage to maintain availability [3].

5.2 Threat Modeling & Requirement Analysis

A threat model to identify possible threats and how they are minimized or mitigated has been done in correspondence to our model. This is done to predict the effectiveness of our model. We put key focus on our model's use of JWT tokens, TLS, and public key cryptography.

5.2.1 Threat Modeling

To identify and categorize potential threats, we use the STRIDE threat model, which addresses different types of security threats as follows:

- **T1. Identity Spoofing:** An attacker could impersonate another user to gain unauthorized access to the Metaverse. This could lead to unauthorized usage of resources and potentially damaging actions performed under the guise of a legitimate user.

- **T2. Data Tampering:** Unauthorized modifications of critical authentication data, such as JWT tokens, could be performed by an attacker for malicious purposes. This might include altering tokens or user credentials to gain unauthorized access or corrupt data integrity.

- **T3. Information Disclosure:** Sensitive data might be inadvertently disclosed to unauthorized parties through vectors such as eavesdropping or improper data handling practices. This could expose user credentials and personal information.

- **T4. Denial of Service (DoS):** The authentication service might be targeted by attacks aimed at disrupting service availability, rendering the authentication service unusable and affecting overall system accessibility.

- **T5. Replay Attacks:** Captured JWT tokens or authentication responses might be reused by an attacker to gain unauthorized access, bypassing authentication checks if not properly mitigated.

- **T6. Repudiation:** Users or entities might deny actions or transactions they conducted in the Metaverse. Without proper non-repudiation mechanisms, it becomes challenging to prove the origin and authenticity of certain actions.

- **T7. Privilege Escalation:** Attackers might exploit vulnerabilities to gain higher access levels within the system, potentially leading to the misuse of system resources.

- **T8. Device Compromise:** An attacker could compromise the user's device, gaining unauthorized access to JWT tokens and other authentication data. Ensuring device integrity is vital to maintaining a secure authentication process.

- **T9. Consent Violations:** User data could be processed or released without explicit user consent, undermining user trust and leading to unauthorized use of personal information.

- **T10. Lack of Data Control:** Users might have insufficient control over how their authentication data is utilized and shared, leading to potential privacy breaches.

- **T11. Side-Channel Attacks:** Attackers might exploit physical characteristics or implementation flaws to extract sensitive information, such as JWT tokens.

- **T12. Phishing:** Users could be tricked into revealing their authentication credentials through deceptive means, leading to significant security breaches.

5.2.2 Requirement Analysis

This section details the functional, security, and privacy requirements that ensure the robustness of the authentication model against identified threats.

Functional Requirements (FR)

- **F1. Secure Session Handling:** The system must securely manage authentication sessions, covering establishment, maintenance, and termination to prevent session hijacking and other related attacks (addresses T4, T5). This includes robust session management protocols to ensure continuous and secure access.

- **F2. Device Integrity Verification:** The system must perform integrity checks on the user's device to confirm it has not been compromised before initiating authentication (addresses T8). This ensures that only uncompromised devices can participate in the authentication process.

- **F3. Multi-Device Accessibility:** Users should seamlessly access the Metaverse across various devices, maintaining session continuity and ease of use. This enhances user experience and ensures consistent access to services.

- **F4. Dynamic Cipher Selection:** The system must support dynamic selection of cryptographic ciphers to ensure the strongest encryption is always used (addresses T2, T3). This adaptability ensures that the system remains secure against evolving threats.

- **F5. Cryptographic Challenge Creation:** The main server must create unique cryptographic challenges for each authentication attempt to ensure authenticity and integrity (addresses T1, T5). This prevents attackers from reusing previous authentication data.

- **F6. User Intent Confirmation:** The system must confirm user intent before proceeding with authentication to prevent unauthorized access due to unattended devices (addresses T1, T12). This could involve multi-factor authentication or user interaction verification.

- **F7. Cryptographic Response Creation:** The user device must create the correct response to the unique challenge sent by the main server. This addresses T1, T3, and T12 in the authentication system.

Security Requirements (SR)

- **S1. Robust Authentication Mechanisms:** Implement public key cryptography-based challenge-response protocols to verify user identities, mitigating identity spoofing (T1). This includes using established algorithms such as RSA and secure key exchange mechanisms.

- **S2. Data Integrity Assurance:** Ensure data integrity by protecting authentication data, such as JWT tokens, from unauthorized alterations (T2). This can be achieved through digital signatures and integrity verification protocols.

- **S3. Encrypted Data Transmission:** Encrypt all critical data transmissions using TLS to maintain confidentiality and prevent information leakage (T3). This ensures that data in transit is protected from eavesdropping and interception.

- **S4. Defense Against DoS Attacks:** Implement protective measures such as rate limiting and anomaly detection to safeguard against DoS attacks (T4). These measures help maintain service availability even under attack.

- **S5. Prevention of Replay Attacks:** Use nonces or timestamps within JWT tokens to protect against replay attacks, ensuring each authentication attempt is unique (T5). This adds a layer of protection against session hijacking.

- **S6. Controlled Access:** Utilize robust access control mechanisms, like role-based access control so that hierarchy is maintained to prevent unauthorized privilege escalation (T7). This ensures that users can only access resources appropriate to their role.

- **S7. Non-Repudiation via Digital Signatures:** Use digital signatures within JWT tokens to ensure actions and transactions cannot be denied, mitigating repudiation threats (T6). Digital signatures provide a cryptographic guarantee of the origin and integrity of messages.

- **S8. Biometric Authentication:** Integrate biometric checks, such as iris scans, to ensure that the authentication process is tied to a physical user, preventing spoofing and unauthorized access (addresses T1, T8). This adds a highly secure and user-friendly authentication factor.

- **S9. Side-Channel Attack Mitigation:** Implement countermeasures against side-channel attacks, such as constant-time algorithms and noise introduction, to protect sensitive information (T11). These measures reduce the risk of information leakage through side channels.

- **S10. Phishing Protection:** Educate users and implement verification steps to prevent phishing attacks, ensuring credentials are not inadvertently disclosed (T12). User awareness programs and technical measures can significantly reduce phishing risks.

Privacy Requirements (PR)

- **P1. User Consent Management:** Ensure that user authentication data, including JWT tokens, is processed only after explicit user consent, addressing consent violations (T9) and data control issues (T10). This can involve user confirmation steps and transparent data handling policies.

- **P2. Data Minimization:** Collect only the minimum necessary data for authentication to reduce exposure and potential misuse of user information (addresses T3, T10). This practice limits the amount of sensitive data processed and stored.

- **P3. Transparent Data Practices:** Inform users about how their data will be used and provide them with control over their personal information (addresses T10). Transparency builds trust and ensures compliance with data protection regulations.

Table 5.1: Threats Mapped to Requirements

Name of Threats	**Functional Requirements**	**Security Requirements**	**Privacy Requirements**
T1 (Identity Spoofing)	F6,F7	S1, S8	
T2 (Data Tampering)	F4	S2	
T3 (Information Disclosure)	F4	S3	P2
T4 (Denial of Service)	F1	S4	
T5 (Replay Attacks)	F1, F5	S5	
T6 (Repudiation)	F7	S7	
T7 (Privilege Escalation)		S6	
T8 (Device Compromise)	F2	S8	
T9 (Consent Violations)			P1
T10 (Lack of Data Control)			P1, P2, P3
T11 (Side-Channel Attacks)		S9	
T12 (Phishing)	F6,F7	S10	

5.2.3 Threats Mapped to Requirements

By fulfilling these functional, security, and privacy requirements, our Federated Metaverse authentication model aims to provide a secure, reliable, and user-friendly authentication system using JWT tokens, TLS, and public key cryptography. The model prioritizes user control over personal information and ensures the integrity and confidentiality of the authentication process, ensuring a safe virtual environment and proving the possible effectiveness of the model. Table 5.1 maps the identified threats to these requirements, highlighting how our model addresses various security concerns.

5.2.4 Security and Functionality Comparison

A comparison of the security features of various proposed authentication protocols in the metaverse is conducted, with the findings summarized in Table 5.2. Panda and Chattopadhyay [14] introduced a mutual authentication protocol that relies on elliptic curve cryptography, aiming to ensure secure communication between IoT devices and cloud servers. They assert that their proposed protocol offers protection against various security threats, encompassing impersonation and replay attacks. Li et al. [21] designed a mutual authentication scheme centered on blockchain for both users and servers. Their approach tackles the Single Point of Failure (SPoF) challenge inherent in centralized authentication systems by advocating for a decentralized authentication model facilitated by blockchain technology. They assert that their scheme provides security against impersonation and man-in-the-middle attacks, alongside offering perfect forward secrecy. However, it lacks coverage of security aspects such as safeguarding against insider threats and ensuring anonymity. Ryu et al. [24] introduced a blockchain-supported authentication protocol for metasystems that leverages elliptic curve cryptography to secure communications between users and platform servers. Despite these features, the protocol falls short in defending against real-world impersonation attacks and securing session keys. Li et al. [9] developed a server-assisted authentication method based on chaotic mapping, but it also fails to mitigate impersonation attacks. Similarly, Zheng et al. [33] proposed an effective session key establishment technique between users using chaotic mapping; however, it is susceptible to data leakage during MITM (Man-In-The-Middle) attacks. According to the Table, our proposed system meets all the security requirements that have been included in the comparison comprehensively.

Table 5.2: A Comparison of Security and Functionality Features

Attack/Feature	**Panda et al. [14]**	**Li et al. [21]**	**Ryu et al. [24]**	**Zheng et al. [33]**	**Li et al. [9]**	**Our Protocol**
Stolen devices attack	–	–	✓	✓	✓	✓
Password guessing attack	✓	–	✓	–	–	✓
Shoulder Surfing	✓	–	✓	–	–	✓
Impersonate attack	✓	✓	✓	–	–	✓
Session Key disclosure attack	✓	✓	✓	×	✓	✓
Perfect forward secrecy	✓	✓	✓	–	–	✓
Reply attack	✓	✓	✓	✓	✓	✓
MITM attack	✓	✓	✓	×	×	✓
Insider attack	✓	–	✓	–	–	✓
Ephemeral secret leakage	×	–	✓	–	–	✓
Mutual authentication	×	×	✓	–	✓	✓
Anonymity	✓	×	✓	–	–	✓
Untraceability	✓	×	✓	–	–	✓
Denial-of-Service(DoS) Attack	×	✓	×	✓	×	✓
Adaptability/Dynamic	×	×	×	×	×	✓
Scalability	×	✓	✓	✓	×	✓

✓ Protocol/scheme is secure or provides functionality feature

× Protocol/scheme is insecure and does not provide functionality feature

- Cannot be considered

Chapter 6

Conclusion & Future Work

6.1 Conclusion

The immersive virtual world termed the Metaverse, offers an exciting new era in human activity and interaction. The criteria of security and authentication become increasingly vital as its realization comes closer and closer every day. This study ventured into the Metaverse, analyzed its particular security issues related to authentication and session management, and proposed innovative solutions. The cornerstone of our approach was the creation of a federated authentication model that implements a dynamic authentication system, adapting to the requirements of both guest and central servers, while adhering to modern information security standards and regulations set by the central server as part of our strategy for securing the Metaverse. The complexities and vulnerabilities inherent in this ground-breaking environment are addressed by this approach, which has been designed to dynamically adapt to the constantly changing digital terrain. The hybrid authentication system acts as a sentinel against potential threats by fusing user-centric design principles, rigorous verification methods, and adaptability to changing conditions focusing on both security and user experience.

6.2 Future Work

Implementation and Scalability Evaluation:
A metaverse, synonymous with second life or virtual world, is a futuristic concept which has not reached the advanced stage of development. Despite the fact that there are attempts by large corporations emanating from Facebook and Apple to germinate the idea there hasn't been a complete realization hitherto. Use of existing concepts made it possible to create an outline for this paper but it's only through realizing them that their feasibility and workability will be measured. This allows for real-world testing and evaluation of the system's scalability and performance within a potentially massive user base of the metaverse. Particular focus should be placed on the feasibility of the Public Key Infrastructure (PKI) for managing a large number of users and digital identities.

Advanced Public Key Cryptography (PKC) Integration:
When it comes to the evolution of cryptography, the study of integrating advanced PKC methods including post-quantum cryptography (PQC) ought to be looked at.

PQC strategies are tailored specifically aimed at preventing hacks that could be mounted on the metaverse with the use of quantum computers that may come up in future.

Decentralized Identity Management with Blockchain and Self-Sovereign Identity (SSI):
The aim of this thesis was to assess Decentralized Identity (DID) solutions using an island-based, but not completely isolated approach, framework based on SSI. DID frameworks give more control of their identities to users that live in the metaverse. One way of improving the autonomy of users and their privacy would be exploring how to merge DID solutions and the intended authentication system.

User Experience (UX) and Privacy Considerations:
It is very important to maintain a proper balance between authority and convenience of users. More ways need to be looked for to enable secure and easy identification with minimal friction while at the same time safeguarding user privacy. One possible direction is to investigate alternative methods of multi-factor authentication (MFA) that would help create an environment where the users feel safe and their information is secure in metaverse.

Bibliography

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] H. Krawczyk, K. G. Paterson, and H. Wee, "Hmac: Keyed-hashing for message authentication," *RFC 2104*, 1997.

[3] A. S. Tanenbaum and M. van Steen, *Distributed systems: Principles and paradigms.* Pearson Prentice Hall, 2007.

[4] O. Boehm, J. Caumanns, M. Franke, and O. Pfaff, "Federated authentication and authorization: A case study," in *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, 2008, pp. 356–362. DOI: 10.1109/EDOC.2008.36.

[5] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," *RFC 5280*, 2008.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63–74, 2008.

[7] T. Z. Suel, "Keyauth: Bringing public-key authentication to the masses," *arXiv.org*, 2012, Accessed: 2012-09-05. [Online]. Available: https://arxiv.org/abs/1209.0967.

[8] P. Fremantle, B. Aziz, J. Kopecký, and P. Scott, "Federated identity and access management for the internet of things," in *2014 International Workshop on Secure Internet of Things*, 2014, pp. 10–17. DOI: 10.1109/SIoT.2014.8.

[9] X. Li, J. Niu, S. Kumari, and S. H. Islam, "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security," *Wireless Personal Communications*, vol. 89, no. 2, Jul. 2016. DOI: 10.1007/s11277-016-3293-x.

[10] E. Rescorla, "The transport layer security (tls) protocol version 1.3," *RFC 8446*, 2018.

[11] C. George, M. Khamis, D. Buschek, and H. Hussmann, "Investigating the third dimension for authentication in immersive virtual reality and in the real world," *IEEE Conference Publication*, Aug. 2019. DOI: 10.1109/VR.2019.8797862.

[12] M. Peng, F. Le, C. Luo, J. Zhang, and S. Du, "Ddos attack detection and defense system based on entropy and fuzzy clustering," *IEEE Access*, vol. 7, pp. 15 554–15 563, 2019.

[13] J. Liebers and S. Schneegass, “Gaze-based authentication in virtual reality,” in *ACM Symposium on Eye Tracking Research and Applications*, 2020. DOI: 10.1145/3379157.3391421.

[14] P. Panda and S. Chattopadhyay, “A secure mutual authentication protocol for iot environment,” *J. Reliable Intell. Environ.*, vol. 6, pp. 79–94, 2020. DOI: 10.1007/978-981-15-2086-8_17.

[15] H. Zhu, W. Jin, M. Xiao, S. Murali, and M. Li, “Blinkey,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–29, 2020. DOI: 10.1145/3432217.

[16] A. Ajgar and Y. m. Ajgar. “Key security challenges for women in metaverse and solutions to overcome those challenges.” (Sep. 2022).

[17] X. Bao, M. Shou, and J. Yu. “Icis 2022 proceedings: Exploring metaverse: Affordances and risks for potential users.” (2022).

[18] Y. Canbay, A. Utku, and P. Canbay, “Metaverse: The potential threats in the virtual world,” Oct. 2022.

[19] Y. Chow, W. Susilo, Y. Li, N. Li, and N. Chau, “Visualization and cybersecurity in the metaverse: A survey,” *Journal of Imaging*, vol. 9, no. 1, p. 11, 2022. DOI: 10.3390/jimaging9010011.

[20] P. Kürtünlüoğlu, B. Akdik, and E. Karaarslan. “Security of virtual reality authentication methods in metaverse: An overview.” (2022).

[21] Y. Li, M. Xu, and G. Xu, “Blockchain-based mutual authentication protocol without ca,” *The Journal of Supercomputing*, vol. 78, no. 15, pp. 17 261–17 283, Oct. 2022. DOI: 10.1007/s11227-022-04558-5.

[22] Z. Lv, L. Qiao, Y. Li, Y. Yuan, and F. Wang, “Blocknet: Beyond reliable spatial digital twins to parallel metaverse,” *Patterns*, vol. 3, no. 5, p. 100 468, 2022. DOI: 10.1016/j.patter.2022.100468.

[23] R. Miller, N. K. Banerjee, and S. Banerjee, “Within-system and cross-system behavior-based biometric authentication in virtual reality,” in *IEEE Conference Publication*, May 2022. DOI: 10.1109/VR50857.2022.9090572.

[24] J. Ryu, S. Son, J. Lee, Y. Park, and Y. Park, “Design of secure mutual authentication scheme for metaverse environments using blockchain,” *IEEE Access*, vol. 10, pp. 98 944–98 958, 2022. DOI: 10.1109/ACCESS.2022.3206457.

[25] Y. Wang, Z. Su, N. Zhang, *et al.*, “A survey on metaverse: Fundamentals, security, and privacy,” Sep. 2022.

[26] H. Xu, Z. Li, Z. Li, X. Zhang, Y. Sun, and L. Zhang. “Metaverse native communication: A blockchain and spectrum prospective.” (Mar. 2022).

[27] E. Carle, “Ask a techspert: What are passkeys?” *Google*, 2023, Accessed: 2023-10-10. [Online]. Available: https://blog.google/inside-google/googlers/ask-a-techspert/how-passkeys-work/.

[28] CSO Online, “The metaverse brings a new breed of threats to challenge privacy and security gatekeepers,” *CSO Online*, 2023. [Online]. Available: https://www.csoonline.com/article/574383/the-metaverse-brings-a-new-breed-of-threats-to-challenge-privacy-and-security-gatekeepers.html.

[29] M. S. Ferdous, A. Ionita, and W. Prinz, "Ssi4web: A self-sovereign identity (ssi) framework for the web," in *Proceedings of the Conference*, BRAC University, 66 Mohakhali, 1212 Dhaka, Bangladesh, Fraunhofer Institute for Applied Information Technology, Schloss Birlinghoven, 53757 Sankt Augustin, Germany, 2023.

[30] H. Fereidouni, O. Fadeitcheva, and M. Zalai, "Iot and man-in-the-middle attacks," Aug. 2023.

[31] G. Garrido, V. Nair, and D. Song. "Going incognito in the metaverse." (May 2023).

[32] K. Kim, J. Ryu, H. Lee, Y. Lee, and D. Won, "Distributed and federated authentication schemes based on updatable smart contracts," *Electronics*, vol. 12, no. 5, p. 1217, 2023. DOI: 10.3390/electronics12051217. [Online]. Available: https://doi.org/10.3390/electronics12051217.

[33] C. Lai, H. Wang, H. Ma, and D. Zheng, "Blockchain-based multi-factor group authentication in metaverse," in *2023 IEEE/CIC International Conference on Communications in China (ICCC)*, Dalian, China, 2023, pp. 1–6. DOI: 10.1109/ICCC57788.2023.10233657.

[34] S. C. Sethuraman, A. R. Mitra, A. Ghosh, G. Galada, and S. Anitha. "Metasecure: A passwordless authentication for the metaverse." (2023).

[35] K. Yang, Z. Zhang, T. Youliang, and J. Ma, "A secure authentication framework to guarantee the traceability of avatars in metaverse," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3817–3832, 2023. DOI: 10.1109/tifs.2023.3288689.