Blockchain-Enabled Secure Communication for Unmanned Aerial Vehicle (UAV) Networks

Randhir Kumar
Department of Electrical Engineering,
Indian Institute of Technology,
Hyderabad
Telangna 502285, India
randhir.honeywell@ieee.org

Ahamed Aljuhani
Department of Computer
Engineering, Faculty of Computers
and Information Technology,
University of Tabuk
Tabuk 47512, Saudi Arabia
A_aljuhani@ut.edu.sa

Prabhat Kumar
Department of Software Engineering,
LUT School of Engineering Science,
LUT University
Lappeenranta 53850, Finland
prabhat.kumar@lut.fi

Abhinav Kumar
Department of Electrical Engineering,
Indian Institute of Technology,
Hyderabad
Telangna 502285, India
abhinavkumar@ee.iith.ac.in

Antony Franklin
Department of Computer Science
Engineering, Indian Institute of
Technology, Hyderabad
Telangna 502285, India
abhinavkumar@ee.iith.ac.in

Alireza Jolfaei
College of Science and Engineering,
Flinders University
Adelaide, Australia
alireza.jolfaei@flinders.edu.au

ABSTRACT

While 5G can provide high-speed Internet connectivity and over-the-horizon control for Unmanned Aerial Vehicles (UAVs), authentication becomes a key security component in 5G-enabled UAVs. This is due to fact that the communicating entities in the network mostly uses unsecured communication channel to exchange critical surveillance data. Authentication thus plays a crucial role in the 5G-enabled UAV network, providing a range of security services such as credential privacy, Session-Key (SK) security, and secure mutual authentication. However, transparency, anonymity, traceability and centralized control are few major security requirements that cannot be fulfilled by the traditional authentication schemes. One of the upcoming technologies that can provide a solution for present centralized 5G-enabled UAV network is blockchain-based authentication scheme. Motivated from aforementioned discussion, this paper presents a Permissioned Blockchain empowered Secure Authentication and Key Agreement framework in 5G-enabled UAVs. In this framework, first an authentication phase between UAV-to-UAV, UAV-to-Edge Server (ES) and Edge-to-Cloud Server (CS) supporting mutual authentication and key agreement is proposed. The authenticated surveillance data collected from UAV is used by the peer-to-peer CS for transaction verification, block creation and addition using smart contract-based consensus mechanism. The practical implementation of framework shows the effectiveness of the proposed approach.

DroneCom '22, October 17, 2022, Sydney, NSW, Australia
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9514-4/22/10…$15.00
https://doi.org/10.1145/3555661.3560861

CCS CONCEPTS

• **Security and privacy → Symmetric cryptography and hash functions**.

KEYWORDS

5G, Authentication and Key Agreement, Blockchain, Unmanned Aerial Vehicles

ACM Reference Format:
Randhir Kumar, Ahamed Aljuhani, Prabhat Kumar, Abhinav Kumar, Antony Franklin, and Alireza Jolfaei. 2018. Blockchain-Enabled Secure Communication for Unmanned Aerial Vehicle (UAV) Networks. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3555661.3560861

1 INTRODUCTION

Fifth generation (5G) networks resulted from the rapid growth of emerging technologies and applications, such as mobile edge computing, Internet of Things (IoT), and unmanned aerial vehicles (UAVs) [7]; such technologies require reliability, low latency, and high data rates. Compared to 4G networks, 5G networks significantly improve key features of sensing technologies. For example, they offer higher data rates, delivering more than 10 Gbps; they increase coverage area for wireless communications; and they provide low-latency operations [13]. The integration of 5G networks with innovative technologies, such as software-defined networking, network functions virtualization, fog computing, IoT, and UAVs, significantly improves end-user quality of service, and machine-to-machine and human-to-human communications [14].

As UAVs have grown and rapidly developed in the past decade to meet market growth, the 5G-enabled UAV can achieve superior performance in terms of latency, capacity, and communication coverage [10]. The 5G-enabled UAV can be deployed in a diverse range of industrial applications, including smart agriculture, smart health care, and smart transportation systems, to provide a cost-effective, flexible, and more efficient platform for end users. Because of the nature of UAV operations, which rely on continuous data exchange with other sensing platforms, the security and privacy of the information exchanged between devices remains a significant challenge. UAV drones communicate with different sensing and smart devices, receiving and sending critical data to the ground station server (GSS). However, maintaining the security and privacy of the communication among smart devices, drones, and GSSs remains a significant issue [3]. Insecure communication among different sensing technologies, on the contrary, might be exploited by intruders and pose significant security risks. An attack, such as man-in-the-middle, intercepts communication between smart devices with the goal of capturing, modifying, and forwarding messages so that authentic participants believe the messages came from an authentic device. Another threat to 5G-enabled UAVs is distributed denial-of-service attacks, in which an attacker attempts to make smart devices unavailable or slow in responding to authentic participants [1].

Recent technologies, such as blockchain, attracted a great deal of attention recently owing to the numerous benefits that such a technologies offer. Blockchain technology follows decentralized, distributed, and peer-to-peer (P2P) communication networks, with data stored on each node to ensure the data integrity and confidentiality of all transactions [9]. Blockchain technology has been integrated with various emerging technologies and deployed in a variety of critical domains because it provides effective, robust, and more secure solutions against different types of cyberattacks [5, 12]. Precisely, blockchain-assisted authentication for 5G-enabled UAVs has been adopted to certify drone identities and ensure secure data-sharing communication in integrated environments, such as fog and cloud [2]. Although several studies on blockchain-enabled authentication for UAVs have been conducted to ensure the data integrity and confidentiality of all transactions in such integrated environments, some major challenges remain [8, 11]. Existing solutions, for example, lack cost-effective, efficient, and scalable blockchain-enabled authentication for UAV systems. Blockchain is deeply integrated with other sensing technologies and applications, such as IoT, fog, and cloud; however, it also lacks full integration with UAV systems.

Motivated by the aforementioned discussion, this article introduces a "UAV Communication", an architecture for secure authentication and key agreement using permissioned blockchain in 5G-enabled UAVs. Specifically, an authentication scheme is designed during the hop-by-hop device data relay between any two UAVs that are forwarding the data, other authentication scheme is designed when a UAV forwards its data to the associated edge server and final authentication scheme is designed when a edge server forwards its data to the cloud server. This is done specifically to ensure that the data exchanged between the components comes from the expected reliable source. The edge server encrypts the surveillance data before sending it to a cloud server using the cloud server's public key. The edge server is also utilized to make the decision of which transactions in the received block should be encrypted and which can be kept unencrypted in order to create the hybrid blockchain. A distributed cloud centre with smart contract facility makes up the cloud server. The cloud servers initially employ smart contracts to validate the received transaction. The cloud server generates the block once it has been verified, works with other cloud servers to run a Proof-of-Authority (round-based aura algorithm) consensus method, and finally adds the entire block to the blockchain.

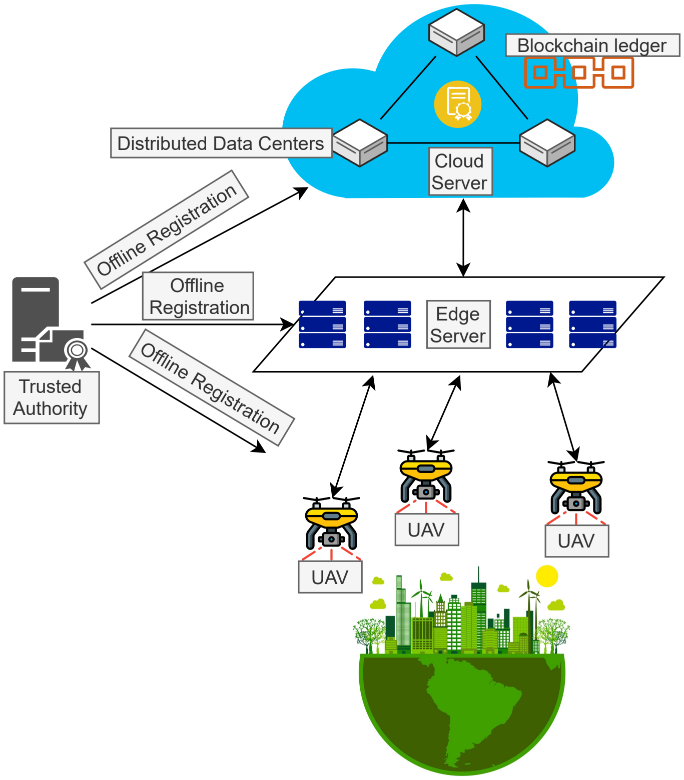


Figure 1: Network Model for proposed UAV Network framework

2 SYSTEM MODELS

In this section, we present the network and a threat model that is used to design proposed framework. Both models are explained below:

2.1 Network Model

The network model for the proposed framework is illustrated in Fig. 1. The framework demonstrates the 5G-enabled UAV network where UAVs are responsible to collect data from IoT smart devices placed in various location of smart cities. The proposed framework has a Trusted Authority ($\mathcal{TA}$), that is considered as a fully trusted entity and is responsible for registering all UAVs, edge servers and cloud servers prior to their deployment. In this strategy, the UAV gathers surveillance data and securely transmits it to edge servers after establishing a session key between them. The encrypted data is relayed to cloud server on a hop-by-hop forwarding basis from edge servers. The cloud server has distributed data centers forming peer-to-peer cloud server network. The cloud data center uses its private key to decrypt the received data. The smart contract-based consensus algorithm is used by data centers to construct, verify and then will add transaction to blockchain ledger. The authenticated data stored in blockchain is temper proof and is rescued from data poisoning attacks.

2.2 Threat Model

The "Dolev-Yao", often known as the DY model, is the first threat model that we employ in this paper [6]. According to this model, an adversary known as $\mathcal{A}$ can not only intercept, alter, or delete communication messages between any two participants, but can also add harmful messages to the channel. It is assumed that Trusted Authority ($\mathcal{TA}$) is a perfectly reliable entity. Unmanned Aerial Vehicles (UAV) is regarded as untrusted entity, whereas edge and cloud servers (CS) are regarded as semi-trusted entities. The Canetti and Krawczyk adversary model (often referred to as CK-adversary) is also used as a further threat model [4]. In this instance, an adversary $\mathcal{A}$ has the ability to hijack a live session between two network users by stealing their secret credentials and the session key/state.

3 THE PROPOSED FRAMEWORK

3.1 Proposed Authentication and Key Agreement Module

3.1.1 Initialization Phase. This phase explores, how Trusted Authority ($\mathcal{TA}$) chooses the parameters to register the entities in proposed framework. The detailed process is discussed below. First, non singular elliptic curve is selected by the $\mathcal{TA}$ i.e., $\mathcal{E}_t(\beta, \gamma)$ $\mathcal{S}^2 = \mathcal{T}^3 + \alpha\mathcal{T} + \gamma$ (mod $\mathcal{W}_n$), where $\mathcal{W}_n$ is a large prime value and $\beta, \gamma \in \mathcal{V}^* = \{1,2,3,\ldots, \mathcal{W}_n\}$ are the two points i.e, infinity point and zero point $\mathcal{ZO}$. Further, the $\mathcal{TA}$ chooses a base point $\mathcal{BP} \in \mathcal{E}_t(\beta, \gamma)$ of order $\mathcal{U}$ as bigger as $\mathcal{W}_n$. Furthermore, $\mathcal{TA}$ chooses a cryptographic hash function i.e.; $\mathcal{HF}(.)$ using SHA-256. In addition, $\mathcal{TA}$ chooses an identity $\mathcal{ID}_{\mathcal{TA}}$, and picks a private key $\mathcal{TA}_{\mathcal{PR}} \in \mathcal{V}^*$ and evaluates a public key $\mathcal{TA}_{\mathcal{PB}} = \mathcal{TA}_{\mathcal{PR}} * \mathcal{BP}$. Finally, the $\mathcal{TA}$ preserve a private key ($\mathcal{TA}_{\mathcal{PR}}$) secret and disseminates public parameters$\{(\mathcal{E}_t(\beta, \gamma), \mathcal{BP}, \mathcal{HF}(.), \mathcal{TA}_{\mathcal{PB}})\}$.

3.1.2 Registration Phase. This phase describes a registration process of each entities and shares the communication parameters. $(a)\mathcal{UAV}Registration$: The $\mathcal{TA}$ registers a UAV nodes $\mathcal{UAV}$, where $\mathcal{UAV}=\{1,2,\ldots, \mathcal{UAV}\}$

Step-1: The $\mathcal{TA}$ chooses an unique identity $\mathcal{ID}_{\mathcal{UAV}}$ for registration of UAV devices. Further, $\mathcal{TA}$ evaluates a pseudo identity $\mathcal{PSID}_{\mathcal{UAV}D}$= HF($\mathcal{ID}_{\mathcal{UAV}D}$ || ${C_{\mathcal{UAV}}}^{PR}$ || $\mathcal{TSTP}_{\mathcal{UAV}D}$), where $\mathcal{TSTP}_{\mathcal{UAV}D}$ is a registration time of $\mathcal{UAV}$ and generates a certificate $\mathcal{CRT}_{\mathcal{UAV}D} = \mathcal{TA}_{\mathcal{PR}}$ + HF($\mathcal{PSID}_{\mathcal{UAV}D}$ || ${C_{\mathcal{UAV}}}^{PB}$ || $\mathcal{TA}_{\mathcal{PB}}$) * ${C_{\mathcal{UAV}}}^{PR}$ mod ($\mathcal{W}_n$).

Step-2: TA chooses a random number $RN\mathcal{UAV} \in \mathcal{V}^*$, and evaluates a partial private key i.e., $\mathcal{PPR}_{\mathcal{UAV}D}$= HF($\mathcal{TA}_{\mathcal{PR}}$ || ${C_{\mathcal{UAV}}}^{PR}$ || $RN\mathcal{UAV}$), and evaluates a public key $\mathcal{PB}_{\mathcal{UAV}D}$= $\mathcal{PPR}_{\mathcal{UAV}D}$ * $\mathcal{B}$ for each $\mathcal{UAV}$ and preserves registration information ($\mathcal{PSID}_{\mathcal{UAV}D}$, $\mathcal{PPR}_{\mathcal{UAV}D}$, $\mathcal{CRT}_{\mathcal{UAV}D}$) on board units $\mathcal{OBU}_D$ of UAV devices $\mathcal{UAV}$. Finally, $\mathcal{TA}$ deletes a partial private key $\mathcal{PPR}_{\mathcal{UAV}D}$ and disseminates the public key $\mathcal{PB}_{\mathcal{UAV}D}$ for communication.

$(b)\mathcal{ES}Registration$: The $\mathcal{TA}$ registers a ES $\mathcal{ES}_t$, where $\mathcal{ES}_t$=$\{1,2,\ldots, \mathcal{ES}_t\}$

Step-1: The $\mathcal{TA}$ chooses an unique identity $\mathcal{ID}_{\mathcal{ES}}$ for registration of ES. Further, $\mathcal{TA}$ evaluates a pseudo identity $\mathcal{PSID}_{\mathcal{ES}t}$ HF($\mathcal{ID}_{\mathcal{ES}t}$ || ${C_{\mathcal{ES}}}^{PR}$ || $\mathcal{TSTP}_{\mathcal{ES}t}$), where $\mathcal{TSTP}_{\mathcal{ES}t}$ is a registration time $\mathcal{ES}_r$ and produces certificate $\mathcal{CRT}_{\mathcal{ES}t} = \mathcal{TA}_{\mathcal{PR}}$ + HF($\mathcal{PSID}_{\mathcal{ES}t}$ || ${C_{\mathcal{ES}}}^{PB}$ || $\mathcal{TA}_{\mathcal{PB}}$) * ${C_{\mathcal{ES}}}^{PR}$ mod ($\mathcal{W}_n$).

Step-2: TA chooses a random number $RN\mathcal{ES}_t \in \mathcal{V}^*$, and evaluates a partial private key $\mathcal{PPR}_{\mathcal{ES}t}$= HF($\mathcal{TA}_{\mathcal{PR}}$ || ${C_{\mathcal{ES}}}^{PR}$ || $RN\mathcal{ES}_t$), and evaluates a public key $\mathcal{PB}_{\mathcal{ES}t}$= $\mathcal{PPR}_{\mathcal{ES}t}$ * $\mathcal{B}$ for each $\mathcal{ES}_t$ preserves a registration information ($\mathcal{PSID}_{\mathcal{ES}t}$, $\mathcal{PPR}_{\mathcal{ES}t}$, $\mathcal{CRT}_{\mathcal{ES}t}$) on its borad unit $\mathcal{OBU}_t$ of ES $\mathcal{ES}_t$. Finally, $\mathcal{TA}$ deletes partial private key $\mathcal{PPR}_{\mathcal{ES}t}$ and disseminates a public key $\mathcal{PB}_{\mathcal{ES}t}$ for communication. The detail of ES i.e., ($\mathcal{CRT}_{\mathcal{ES}t}$, $\mathcal{PB}_{\mathcal{ES}t}$, $\mathcal{PSID}_{\mathcal{ES}t}$, $\mathcal{ID}_{\mathcal{ES}t}$) are disseminated to cloud server $\mathcal{CSV}_l$.

$(c)\mathcal{CS}Registration$: The $\mathcal{TA}$ register cloud servers $\mathcal{CSV}_l$, where $\mathcal{CSV}_l$=$\{1,2,\ldots, \mathcal{CSV}_l\}$.

Step-1: The $\mathcal{TA}$ chooses an unique identity $\mathcal{ID}_{\mathcal{CS}}$ for individual $\mathcal{CSV}_l$ registration. Next, $\mathcal{TA}$ it evaluates pseudo identity $\mathcal{PSID}_{\mathcal{CS}l}$ HF($\mathcal{ID}_{\mathcal{CS}l}$ || ${C_{\mathcal{CS}}}^{PR}$ || $\mathcal{TSTP}_{\mathcal{CS}l}$), where $\mathcal{TSTP}_{\mathcal{CS}l}$ is a time of registration of cloud severs $\mathcal{CSV}_l$ and produces a certificate $\mathcal{CRT}_{\mathcal{CS}l} = \mathcal{TA}_{\mathcal{PR}}$ + HF($\mathcal{PSID}_{\mathcal{CS}l}$ || ${C_{\mathcal{CS}}}^{PB}$ || $\mathcal{TA}_{\mathcal{PB}}$) * ${C_{\mathcal{CS}}}^{PR}$ mod ($\mathcal{W}_n$).

Step-2: A random number is chosen by TA i.e., $RN\mathcal{CSV}_l \in \mathcal{V}^*$, and evaluates a partial private key ie., $\mathcal{PPR}_{\mathcal{CS}l}$= HF($\mathcal{TA}_{\mathcal{PR}}$ || ${C_{\mathcal{CS}}}^{PR}$ || $RN\mathcal{CSV}_l$), and evaluates a public key $\mathcal{PB}_{\mathcal{CS}l}$= $\mathcal{PPR}_{\mathcal{CS}l}$ * $\mathcal{B}$ for each $\mathcal{CSV}_l$ and preserves a registration information ($\mathcal{PSID}_{\mathcal{CS}l}$, $\mathcal{PPR}_{\mathcal{CS}l}$, $\mathcal{CRT}_{\mathcal{CS}l}$) on its borad unit $\mathcal{OBU}_l$ of Cloud $\mathcal{CSV}_l$. Finally, $\mathcal{TA}$ deletes a partial private key $\mathcal{PPR}_{\mathcal{CS}l}$ and disseminates a public key $\mathcal{PB}_{\mathcal{CS}l}$ for communication.

3.1.3 Key Agreement and Authentication Phase. We have discussed various steps used in key agreement and authentication. *(i) UAV to UAV Authentication*

Step-1: $\mathcal{UAV}_1$ picks an unique random value $dr_1 \in Z_q$ and its current timestamp $\mathcal{TSTP}_1$ and evaluates $\mathcal{L}_1$= h ($\mathcal{PSID}_{\mathcal{UAV}_1}$ || $\mathcal{PPR}_{\mathcal{UAV}_1}$ || dr_1 || $\mathcal{TSTP}_1$). Further, $\mathcal{UAV}_1$ applies encryption as $\mathcal{L}_1$ as $\mathcal{L}_2$= $E_{\mathcal{PB}_{\mathcal{UAV}_2}}(\mathcal{L}_1)$. Furthermore, $\mathcal{UAV}_1$ evalautes the $\mathcal{L}_3$= HF($\mathcal{L}_2$ || $\mathcal{CRT}_{\mathcal{UAV}_1}$ || $\mathcal{PSID}_{\mathcal{UAV}_1}$ || $\mathcal{PPR}_{\mathcal{UAV}_1}$ || $\mathcal{TSTP}_1$) and send request message as $\mathcal{M}_1$=$\{\mathcal{PSID}_{\mathcal{UAV}_1}, \mathcal{PPR}_{\mathcal{UAV}_1}, \mathcal{TSTP}_1, \mathcal{L}_2, \mathcal{L}_3\}$ and transmit to UAV through open channel.

Step-2: after message retrieval $\mathcal{M}_1$ by UAV_2 timestamp gets validated i.e., $\mathcal{TSTP}_1^*$ using | $\mathcal{TSTP}_1^*$- $\mathcal{TSTP}_1$ | $< \Delta T$. If it is valid UAV_2 checks for certificates $\mathcal{CRT}_{\mathcal{UAV}_1}$. $\mathcal{B}$= $\mathcal{PB}_{\mathcal{TA}}$ + HF($\mathcal{PB}_{\mathcal{UAV}_1}$ || $\mathcal{PB}_{\mathcal{TA}}$) if validated successfully, then UAV_2 receives $\mathcal{PSID}_{\mathcal{UAV}_1}$ with respect of $\mathcal{PPR}_{\mathcal{UAV}_1}$ from the databases and evaluates $\mathcal{L}_3^*$= h ($\mathcal{L}_2$ || $\mathcal{PSID}_{\mathcal{UAV}_1}$ || $\mathcal{PPR}_{\mathcal{UAV}_1}$ || $\mathcal{CRT}_{\mathcal{UAV}_1}$) to verify whether $\mathcal{L}_3^*$= $\mathcal{L}_3$. if validated successfully, then UAV_2 applies decryption $\mathcal{L}_2$ as $\mathcal{L}_1$= $\mathcal{D}_{\mathcal{PR}_{\mathcal{UAV}}}(\mathcal{L}_2)$.

Step-3: Next, UAV_2 chooses unique random number $\mathcal{UAV}r_1 \in Z_q$ and maintains the timestamp $\mathcal{TSTP}_2$ and generates temporary identity $\mathcal{PPR}^{new}_{\mathcal{UAV}_1}$ and evaluates $\mathcal{UAV}2_1$= HF($\mathcal{PSID}_{\mathcal{UAV}_1}$ || $\mathcal{PSID}_{\mathcal{UAV}_2}$ || $\mathcal{UAV}r_1$ || $\mathcal{TSTP}_2$) and applies encryption $\mathcal{UAV}2_1$ as $\mathcal{UAV}2_2$ = $E_{\mathcal{PB}_{\mathcal{UAV}_1}}(\mathcal{UAV}2_1)$. Next, UAV_2 ($\mathcal{UAV}_2$) creates session key $\mathcal{SES}_{\mathcal{UAV}_2}$= HF($\mathcal{PPR}^{new}_{\mathcal{UAV}_1}$ || $\mathcal{L}_1$ || $\mathcal{UAV}2_1$ || $\mathcal{TSTP}_1$ || $\mathcal{TSTP}_2$), $\mathcal{PPR}^{*}_{\mathcal{UAV}_1}$ = $\mathcal{PPR}^{new}_{\mathcal{UAV}_1}$ $\oplus$ HF($\mathcal{PSID}_{\mathcal{UAV}_2}$ || $\mathcal{PPR}_{\mathcal{UAV}_1}$ || $\mathcal{TSTP}_2$), and $\mathcal{UAV}2_3$=HF($\mathcal{PPR}^{*}_{\mathcal{UAV}_1}$ || $\mathcal{UAV}2_1$ || $\mathcal{CRT}_{\mathcal{UAV}_2}$ || $\mathcal{PSID}_{\mathcal{UAV}_2}$ || $\mathcal{TSTP}_2$) and sends a reply message $\mathcal{M}_2$= $\{\mathcal{PPR}^{*}_{\mathcal{UAV}_1}, \mathcal{UAV}2_2, \mathcal{UAV}2_2, \mathcal{CRT}_{\mathcal{UAV}_2}, \mathcal{PSID}_{\mathcal{UAV}_2}, \mathcal{TSTP}_2\}$ and transmit to $\mathcal{UAV}_1$ through open channel.

Step-4: once reply message is received ($\mathcal{M}_2$) from UAV then timestamp gets validated $\mathcal{TSTP}_2^*$, $\mathcal{UAV}_1$ checks whether | $\mathcal{TSTP}_2^*$ - $\mathcal{TSTP}_2$ | $< \Delta T$. if validated successfully, $\mathcal{UAV}_1$ checks certificates using $\mathcal{CRT}_{\mathcal{UAV}_2}$. $\mathcal{B}= \mathcal{PB}_{\mathcal{TA}} + \mathrm{HF}(\mathcal{PB}_{\mathcal{UAV}_2} || \mathcal{PB}_{\mathcal{TA}})$. Further, $\mathcal{UAV}_1$ applies decryption $\mathcal{UAV}2_2$ to receive $\mathcal{UAV}2_1= \mathcal{D}_{\mathcal{PR}_{\mathcal{UAV}_1}}(\mathcal{UAV}2_2)$. Furthermore, $\mathcal{UAV}_1$ computes $\mathcal{UAV}2^*_3= \mathrm{HF}(\mathcal{PPR}^*_{\mathcal{UAV}_1} || \mathcal{UAV}2_1 || \mathcal{CRT}_{\mathcal{UAV}_2} || \mathcal{PSID}_{\mathcal{UAV}_2} || \mathcal{TSTP}_2)$ and computes $\mathcal{UAV}2^*_3= \mathcal{UAV}2_3$ then $\mathcal{UAV}_1$ evaluates $\mathcal{PPR}^{new}_{\mathcal{UAV}_1}= \mathcal{PPR}^*_{\mathcal{UAV}_1} \oplus \mathrm{HF}(\mathcal{PSID}_{\mathcal{UAV}_2} || \mathcal{PPR}_{\mathcal{UAV}_1} || \mathcal{TSTP}_2)$ and creates session key $\mathcal{SES}_{\mathcal{UAV}_1}=\mathrm{HF}(\mathcal{PPR}^{new}_{\mathcal{UAV}_1} || \mathcal{L}_1 || \mathcal{UAV}2_1 || \mathcal{TSTP}_1 || \mathcal{TSTP}_2)$ and share to $\mathcal{UAV}_2$. Further, $\mathcal{UAV}_1$ picks a timestamp $\mathcal{TSTP}_3$ and checks for session key verification $\mathcal{SESV}_{\mathcal{UAV}_1}$ through $\mathcal{SESV}_{\mathcal{UAV}_1}=\mathrm{HF}(\mathcal{SES}_{\mathcal{UAV}_1} || \mathcal{TSTP}_3)$ and goes for updation of $\mathcal{PPR}_{\mathcal{UAV}_1}$ and $\mathcal{PPR}^{new}_{\mathcal{UAV}_1}$ into the database. Furthermore, $\mathcal{UAV}_1$ generates acknowledgement receipt $\mathcal{M}_3=\{\mathcal{SESV}_{\mathcal{UAV}_1}, \mathcal{TSTP}_3\}$ and transmit to $\mathcal{UAV}_2$ through open channel.

Step-5: once acknowledgement receipt is received by $\mathcal{UAV}_2$ $\mathcal{M}_3$ checks for timestamp $| \mathcal{TSTP}^*_3 - \mathcal{TSTP}_3 | < \Delta T$. Further, $\mathcal{UAV}_2$ checks $\mathcal{SESV}_{\mathcal{UAV}_1}= \mathrm{HF}\ (\mathcal{SESV}_{\mathcal{UAV}_2} || \mathcal{TSTP}_3)$. If matches successfully, then $\mathcal{UAV}_2$ establishment of mutual the session key $\mathcal{SESV}_{\mathcal{UAV}_1}$ (=$\mathcal{SESV}_{\mathcal{UAV}_2}$) with $\mathcal{UAV}_1$. Furthermore, $\mathcal{UAV}_2$ goes for updation $\mathcal{PPR}_{\mathcal{UAV}_1}$ and $\mathcal{PPR}^{new}_{\mathcal{UAV}_1}$ into the database.

(ii) UAV to Edge Server (ES) Authentication

Step-1: $\mathcal{UAV}$ picks an unique random value $dr_1 \in Z_q$ and its current timestamp $\mathcal{TSTP}_1$ and evaluates $\mathcal{L}_1= \mathrm{HF}\ (\mathcal{PSID}_{\mathcal{UAV}} || \mathcal{PPR}_{\mathcal{UAV}} || dr_1 || \mathcal{TSTP}_1)$. Further, $\mathcal{UAV}$ applies $\mathcal{L}_1$ as $\mathcal{L}_2= E_{\mathcal{PB}_{ES}}(\mathcal{L}_1)$. Furthermore, $\mathcal{UAV}$ evaluates $\mathcal{L}_3= \mathrm{HF}(\mathcal{L}_2 || \mathcal{CRT}_{\mathcal{UAV}} || \mathcal{PSID}_{\mathcal{UAV}} || \mathcal{PPR}_{\mathcal{UAV}} || \mathcal{TSTP}_1)$ and creates request message $\mathcal{M}_1=\{\mathcal{PSID}_{\mathcal{UAV}}, \mathcal{PPR}_{\mathcal{UAV}}, \mathcal{TSTP}_1, \mathcal{L}_2, \mathcal{L}_3\}$ and transmit to ES through open channel.

Step-2: after successful receiving of message $\mathcal{M}_1$ timestamp gets validated $\mathcal{TSTP}^*_1$ by ES, $| \mathcal{TSTP}^*_1 - \mathcal{TSTP}_1 | < \Delta T$. If matches successfully, ES checks for certificates $\mathcal{CRT}_{\mathcal{UAV}}$. $\mathcal{B}= \mathcal{PB}_{\mathcal{TA}} + \mathrm{HF}(\mathcal{PB}_{\mathcal{UAV}} || \mathcal{PB}_{\mathcal{TA}})$ if matches successfully, ES receives $\mathcal{PSID}_{\mathcal{UAV}}$ with respect to $\mathcal{PPR}_{\mathcal{UAV}}$ from the database and evaluates $\mathcal{L}^*_3= \mathrm{HF}\ (\mathcal{L}_2 || \mathcal{PSID}_{\mathcal{UAV}} || \mathcal{PPR}_{\mathcal{UAV}} || \mathcal{CRT}_{\mathcal{UAV}})$ to verify whether $\mathcal{L}^*_3= \mathcal{L}_3$. if validated successfully, then ES applies decryption $\mathcal{L}_2$ as $\mathcal{L}_1= \mathcal{D}_{\mathcal{PR}_{\mathcal{ES}}}(\mathcal{L}_2)$.

Step-3: Further, ES picks for unique random number $\mathcal{ES}r_1 \in Z_q$ and its timestamp $\mathcal{TSTP}_2$ and generates temporary identity $\mathcal{PPR}^{new}_{\mathcal{UAV}}$ and evalautes $\mathcal{ES}_1= \mathrm{HF}(\mathcal{PSID}_{\mathcal{UAV}} || \mathcal{PSID}_{\mathcal{ES}} || \mathcal{ES}r_1 || \mathcal{TSTP}_2)$ and applies encryption $\mathcal{ES}_1$ as $\mathcal{ES}_2= E_{\mathcal{PB}_{\mathcal{UAV}}}(\mathcal{ES}_1)$. Furthermore, ES ($\mathcal{ES}$) evaluates session key $\mathcal{SES}_{\mathcal{ES}}= \mathrm{HF}(\mathcal{PPR}^{new}_{\mathcal{UAV}} || \mathcal{L}_1 || \mathcal{ES}_1 || \mathcal{TSTP}_1 || \mathcal{TSTP}_2)$, $\mathcal{PPR}^*_{\mathcal{UAV}}= \mathcal{PPR}^{new}_{\mathcal{UAV}} \oplus \mathrm{HF}(\mathcal{PSID}_{\mathcal{ES}} || \mathcal{PPR}_{\mathcal{UAV}} || \mathcal{TSTP}_2)$, and $\mathcal{ES}_3=\mathrm{HF}(\mathcal{PPR}^*_{\mathcal{UAV}} || \mathcal{ES}_1 || \mathcal{CRT}_{\mathcal{ES}} || \mathcal{PSID}_{\mathcal{ES}} || \mathcal{TSTP}_2)$ and generates reply message $\mathcal{M}_2= \{\mathcal{PPR}^*_{\mathcal{UAV}}, \mathcal{ES}_2, \mathcal{ES}_2, \mathcal{CRT}_{\mathcal{ES}}, \mathcal{PSID}_{\mathcal{ES}}, \mathcal{TSTP}_2\}$ and transmit to $\mathcal{UAV}$ through open channel.

Step-4: once reply messages is successfully recieved ($\mathcal{M}_2$) from ES then timestamp gets validated $\mathcal{TSTP}^*_2$ by $\mathcal{UAV}$ $| \mathcal{TSTP}^*_2 - \mathcal{TSTP}_2 | < \Delta T$. if validated successfully, then $\mathcal{UAV}$ it checks certificates $\mathcal{CRT}_{\mathcal{ES}}$. $\mathcal{B}= \mathcal{PB}_{\mathcal{TA}} + \mathrm{HF}(\mathcal{PB}_{\mathcal{ES}} || \mathcal{PB}_{\mathcal{TA}})$. Further, $\mathcal{UAV}$ applies decryption $\mathcal{ES}_2$ to receives $\mathcal{ES}_1= \mathcal{D}_{\mathcal{PR}_{\mathcal{UAV}}}(\mathcal{ES}_2)$. Furthermore, $\mathcal{UAV}$ evaluates $\mathcal{ES}^*_3= \mathrm{HF}(\mathcal{PPR}^*_{\mathcal{UAV}} || \mathcal{ES}_1 || \mathcal{CRT}_{\mathcal{ES}} || \mathcal{PSID}_{\mathcal{ES}} || \mathcal{TSTP}_2)$ and finds $\mathcal{ES}^*_3= \mathcal{ES}_3$ then $\mathcal{UAV}$. Further, partial private key gets validated $\mathcal{PPR}^{new}_{\mathcal{UAV}}= \mathcal{PPR}^*_{\mathcal{UAV}} \oplus \mathrm{HF}(\mathcal{PSID}_{\mathcal{ES}} || \mathcal{PPR}_{\mathcal{UAV}} || \mathcal{TSTP}_2)$ and evaluates a session key $\mathcal{SES}_{\mathcal{UAV}}=\mathrm{HF}(\mathcal{PPR}^{new}_{\mathcal{UAV}} || \mathcal{L}_1 || \mathcal{ES}_1 || \mathcal{TSTP}_1 || \mathcal{TSTP}_2)$ and share to $\mathcal{ES}$. Furthermore, $\mathcal{UAV}$ picks a current timestamp $\mathcal{TSTP}_3$ and evalaues session key using verification $\mathcal{SESV}_{\mathcal{UAV}}$ i.e, $\mathcal{SESV}_{\mathcal{UAV}}=\mathrm{HF}(\mathcal{SES}_{\mathcal{UAV}} || \mathcal{TSTP}_3)$ and goes for updation of $\mathcal{PPR}_{\mathcal{UAV}}$ and $\mathcal{PPR}^{new}_{\mathcal{UAV}}$ into the database. Furthermore, $\mathcal{UAV}$ generates acknowledgment message $\mathcal{M}_3=\{\mathcal{SESV}_{\mathcal{UAV}}, \mathcal{TSTP}_3\}$ and transmit to $\mathcal{ES}$ through open channel.

Step-5: once acknowledgement reply received successfully, then, $\mathcal{M}_3$ timestamp gets validated i.e., $\mathcal{TSTP}^*_3$ by $\mathcal{ES}$ $| \mathcal{TSTP}^*_3 - \mathcal{TSTP}_3 | < \Delta T$. Further, $\mathcal{ES}$ verification is applied $\mathcal{SESV}_{\mathcal{UAV}}= \mathrm{h}\ (\mathcal{SESV}_{\mathcal{ES}} || \mathcal{TSTP}_3)$. If matches successfully, the $\mathcal{ES}$ ensues establishment of session key $\mathcal{SESV}_{\mathcal{UAV}}$ (=$\mathcal{SESV}_{\mathcal{ES}}$) to $\mathcal{UAV}$. Furthermore, $\mathcal{ES}$ goes for updation of $\mathcal{PPR}_{\mathcal{UAV}}$ and $\mathcal{PPR}^{new}_{\mathcal{UAV}}$ into the database securely.

(iii) Edge Server nodes to Cloud Server Authentication

Step-1: $\mathcal{ES}$ picks an unique random values $dr_1 \in Z_q$ and records timestamp $\mathcal{TSTP}_1$ and evaluates $\mathcal{L}_1= \mathrm{h}\ (\mathcal{PSID}_{\mathcal{ES}} || \mathcal{PPR}_{\mathcal{ES}} || dr_1 || \mathcal{TSTP}_1)$. Further, $\mathcal{ES}$ applies encryption $\mathcal{L}_1$ as $\mathcal{L}_2= E_{\mathcal{PB}_{CS}}(\mathcal{L}_1)$. Furthermore, $\mathcal{ES}$ evaluates the $\mathcal{L}_3= \mathrm{HF}(\mathcal{L}_2 || \mathcal{CRT}_{\mathcal{ES}} || \mathcal{PSID}_{\mathcal{ES}} || \mathcal{PPR}_{\mathcal{ES}} || \mathcal{TSTP}_1)$ and creates a request message $\mathcal{M}_1=\{\mathcal{PSID}_{\mathcal{ES}}, \mathcal{PPR}_{\mathcal{ES}}, \mathcal{TSTP}_1, \mathcal{L}_2, \mathcal{L}_3\}$ and transmit to CS through open channel.

Step-2: after successful receive of message $\mathcal{M}_1$ timestamp gates validated $\mathcal{TSTP}^*_1$ by CS $| \mathcal{TSTP}^*_1 - \mathcal{TSTP}_1 | < \Delta T$. if matches successfully, then CS checks for certificates i.e., $\mathcal{CRT}_{\mathcal{ES}}$. $\mathcal{B}= \mathcal{PB}_{\mathcal{TA}} + \mathrm{HF}(\mathcal{PB}_{\mathcal{ES}} || \mathcal{PB}_{\mathcal{TA}})$ matches successfully, then CS receives $\mathcal{PSID}_{\mathcal{ES}}$ with respect to $\mathcal{PPR}_{\mathcal{ES}}$ from the database and evaluates $\mathcal{L}^*_3= \mathrm{h}\ (\mathcal{L}_2 || \mathcal{PSID}_{\mathcal{ES}} || \mathcal{PPR}_{\mathcal{ES}} || \mathcal{CRT}_{\mathcal{ES}})$ to verify whether $\mathcal{L}^*_3= \mathcal{L}_3$. If matches successfully, then CS applies decryption $\mathcal{L}_2$ as $\mathcal{L}_1= \mathcal{D}_{\mathcal{PR}_{\mathcal{CS}}}(\mathcal{L}_2)$.

Step-3: Again, CS picks a unique random number $\mathcal{CS}r_1 \in Z_q$ and records current timestamp $\mathcal{TSTP}_2$ and generates temporary identity $\mathcal{PPR}^{new}_{\mathcal{ES}}$ and evaluates $\mathcal{CS}_1= \mathrm{HF}(\mathcal{PSID}_{\mathcal{ES}} || \mathcal{PSID}_{\mathcal{CS}} || \mathcal{CS}r_1 || \mathcal{TSTP}_2)$ and applies encryption $\mathcal{CS}_1$ as $\mathcal{CS}_2= E_{\mathcal{PB}_{\mathcal{ES}}}(\mathcal{CS}_1)$. Next, CS ($\mathcal{CS}$) evaluates a session key $\mathcal{SES}_{\mathcal{CS}}= \mathrm{HF}(\mathcal{PPR}^{new}_{\mathcal{ES}} || \mathcal{L}_1 || \mathcal{CS}_1 || \mathcal{TSTP}_1 || \mathcal{TSTP}_2)$, $\mathcal{PPR}^*_{\mathcal{ES}}= \mathcal{PPR}^{new}_{\mathcal{ES}} \oplus \mathrm{HF}(\mathcal{PSID}_{\mathcal{CS}} || \mathcal{PPR}_{\mathcal{ES}} || \mathcal{TSTP}_2)$, and $\mathcal{CS}_3=\mathrm{HF}(\mathcal{PPR}^*_{\mathcal{ES}} || \mathcal{CS}_1 || \mathcal{CRT}_{\mathcal{CS}} || \mathcal{PSID}_{\mathcal{CS}} || \mathcal{TSTP}_2)$ and creates a reply message $\mathcal{M}_2= \{\mathcal{PPR}^*_{\mathcal{ES}}, \mathcal{CS}_2, \mathcal{CS}_2, \mathcal{CRT}_{\mathcal{CS}}, \mathcal{PSID}_{\mathcal{CS}}, \mathcal{TSTP}_2\}$ and transmit to $\mathcal{ES}$ through open channel.

Step-4: once reply message is successfully received ($\mathcal{M}_2$) from CS then timestamp gets validated $\mathcal{TSTP}^*_2$ by $\mathcal{ES}$ i.e., $| \mathcal{TSTP}^*_2 - \mathcal{TSTP}_2 | < \Delta T$. if validated successfully, then $\mathcal{ES}$ checks for certificate i.e., $\mathcal{CRT}_{\mathcal{CS}}$. $\mathcal{B}= \mathcal{PB}_{\mathcal{TA}} + \mathrm{HF}(\mathcal{PB}_{\mathcal{CS}} || \mathcal{PB}_{\mathcal{TA}})$. Further, $\mathcal{ES}$ applies decryption $\mathcal{CS}_2$ to receive $\mathcal{CS}_1= \mathcal{D}_{\mathcal{PR}_{\mathcal{ES}}}(\mathcal{CS}_2)$. Furthermore, $\mathcal{ES}$ evaluates $\mathcal{CS}^*_3= \mathrm{HF}(\mathcal{PPR}^*_{\mathcal{ES}} || \mathcal{CS}_1 || \mathcal{CRT}_{\mathcal{CS}} || \mathcal{PSID}_{\mathcal{CS}} || \mathcal{TSTP}_2)$ and verifies $\mathcal{CS}^*_3= \mathcal{CS}_3$ then $\mathcal{ES}$ evaluates $\mathcal{PPR}^{new}_{\mathcal{ES}}= \mathcal{PPR}^*_{\mathcal{ES}} \oplus \mathrm{HF}(\mathcal{PSID}_{\mathcal{CS}} || \mathcal{PPR}_{\mathcal{ES}} || \mathcal{TSTP}_2)$ and creates a session key $\mathcal{SES}_{\mathcal{ES}}=\mathrm{HF}(\mathcal{PPR}^{new}_{\mathcal{ES}} || \mathcal{L}_1 || \mathcal{CS}_1 || \mathcal{TSTP}_1 || \mathcal{TSTP}_2)$ and share to $\mathcal{CS}$. Further, $\mathcal{ES}$ picks a current timestamp$\mathcal{TSTP}_3$ and applies verification over session key $\mathcal{SESV}_{\mathcal{ES}}$ through $\mathcal{SESV}_{\mathcal{ES}}=\mathrm{HF}(\mathcal{SES}_{\mathcal{ES}} || \mathcal{TSTP}_3)$ and goes for updation of $\mathcal{PPR}_{\mathcal{ES}}$ and $\mathcal{PPR}^{new}_{\mathcal{ES}}$ into the database. Furthermore, $\mathcal{ES}$ generates acknowledgment message $\mathcal{M}_3=\{\mathcal{SESV}_{\mathcal{ES}}, \mathcal{TSTP}_3\}$ and transmit to $\mathcal{CS}$ through open channel.

Step-5: once reply acknowledgement received successfully, $\mathcal{M}_3$ then timestamp gets validated i.e., $\mathcal{TSTP}^*_3$ by $\mathcal{CS}$ $| \mathcal{TSTP}^*_3 - \mathcal{TSTP}_3 | < \Delta T$. Further, $\mathcal{CS}$ verification is applied over $\mathcal{SESV}_{\mathcal{ES}}= \mathrm{h}\ (\mathcal{SESV}_{\mathcal{CS}} || \mathcal{TSTP}_3)$. If matches successfully, then $\mathcal{CS}$ ensures establishment of session key $\mathcal{SESV}_{\mathcal{ES}}$ (=$\mathcal{SESV}_{\mathcal{CS}}$) with $\mathcal{ES}$. Next, $\mathcal{CS}$ goes for updation of $\mathcal{PPR}_{\mathcal{ES}}$ and $\mathcal{PPR}^{new}_{\mathcal{ES}}$ into the database securely.

3.2 Proposed Permissioned Blockchain-based Transaction Writing Module

This phase describes the block verification and addition. The authorized UAV are responsible to create the transactions in the network. Next, the transactions gets verified by the authorized $\mathcal{ES}$ denoted as miner. Further, block gets created and added into the blockchain. The block consists of two parameters namely $\mathcal{V}_i \leftarrow (\mathcal{W}_i, \mathcal{X}_i)$, where $\mathcal{W}_i$ denotes local blockchain ledger of peers where $\mathcal{X}_i$ denotes block pointer. The algorithm consists of four different functions like PROPOSE(), SCORE(), DELIVER(), and ISDECIDED(). The PROPOSE() function is responsible to propose a block with an index in blockchain network. The consensus executed successfully, once block gets decided which is explained in ISDECIDED() function (line number 33). The PROPOSE function uses time-duration (consecutive period) where each miners executes infinite loop and checks *CTS*, in order to propose a block(line number 10). When it gets chance to propose a block (line number 12), a miner sets the parent of block (last block) and does signature (line number 16). Each disseminates() by PROPOSE() function delivered to participating nodes (honest nodes). The DELIVER() functions gets executed (line number 27) and invoked by the honest nodes with correct view of blockchain network. The correct view of the blockchain is made using SCORE() function (line number 23), where the highest blockchain gets highest score. The score is computed where many blockchain, with similar height with last block consists of lowest index wins. This is performed using two different functions i.e.; height and step-num that denotes blockchain height and number of block in the blockchain network. Finally, ISDECIDED() function (line number 33) gets executed when block b is decided after successful consensus, a participating nodes checks for two consecutive rounds i.e., round1 and round2 for block b, in each round block b is mined (verified) by majority of miners ($\mathcal{ES}_l$). Next, block b gets added into the blockchain network after successful mining (verification). The block verification and block Addition is detailed in Algorithm 1.

Algorithm 1 Proof-of-Authority (Round-based Aura Algorithm) for Block Verification and Addition

```
1:  State: $\mathcal{ES} \in \mathcal{ID}_{\mathcal{ES}}$ Set of miners,
2:  $\mathcal{V}_i = (\mathcal{W}_i, \mathcal{X}_i)$ $\mathcal{W}_i$ local blockchain of node $\mathcal{X}_i$ is a DAG of block $\mathcal{W}_i$ and
    pointer $\mathcal{X}_i$
3:  $b \rightarrow$ Block records
4:  $parent \rightarrow$ preceding node of b
5:  $miners \rightarrow$ who mines and sign block b
6:  $step \rightarrow$ new block added to the network
7:  $duration \rightarrow$ each step takes time to validate and added /*miner keep
    proposing a block */
8:  function PROPOSE()$_k$
9:      while True do
10:         sleep ← CTS / duration, CTS → clock time
11:         if $k \in \mathcal{ES}_l \wedge$ step mod $|\mathcal{ES}_l|$ == k then
12:             $b.parent \leftarrow$ lb($\mathcal{V}_i$) , lb → last block
13:             $b.ES \leftarrow \mathcal{X}_i$
14:             $b.step \leftarrow$ step
15:             $\mathcal{V}_i \leftarrow (\mathcal{W}_i \cup$ b, $\mathcal{X}_i \cup b.parent$ )
16: /* disseminate calls DELIVER() function internally*/
17:             disseminate ($\mathcal{V}_i$)
18:             sleep(duration)
19:         end if
20:     end while
21: end function
22: /* returns a score for correct height of blockchain */
23: function SCORE($\mathcal{W}_j$, $\mathcal{X}_j$)
24:     return UNIT256-MAX * height($\mathcal{W}_j$, $\mathcal{X}_j$) - step-num($\mathcal{W}_i$, $\mathcal{X}_i$)
25: end function
26: /*Deliver function computes score with correct height of blockchain */
27: function DELIVER($\mathcal{W}_j$, $\mathcal{X}_j$)
28:     if Score($\mathcal{W}_j$, $\mathcal{X}_j$) > Score($\mathcal{W}_i$, $\mathcal{X}_i$) then
29:         Score($\mathcal{W}_i$, $\mathcal{X}_i$) ← Score($\mathcal{W}_j$, $\mathcal{X}_j$)
30:     end if
31: end function
32: /*it executes when consensus is reached and a block is decided for
    addition in blockchain */
33: function ISDECIDED(b)$_k$
34:     p ← $\mathcal{ID}_{\mathcal{ES}}$
35:     round1 ← (b.step, b.step + p)
36:     round2 ← (b.step + p, b.step + 2*p)
37:     majority1 ← { $b^{'}$ :$b^{'}$ .step ∈ round1}
38:     majority2 ← { $b^{''}$ :$b^{''}$ .step ∈ round2}
39:     return (majority1 ∩ majority2)
40: end function
```

4 SECURITY ANALYSIS

This phase describes security analysis of the proposed model. It includes the formal verification to prevent various attacks. The detailed security analysis is summarized below.

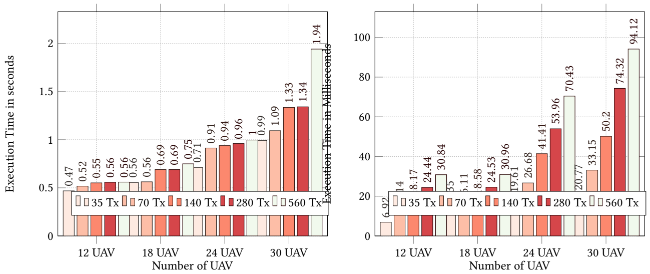
(1) Impersonation Attack: An attacker can generate temporary identity $\mathcal{UAV}$, pseudo identity $\mathcal{PSID}_{\mathcal{UAV}}$, and partial private key $\mathcal{PPR}_{\mathcal{UAV}}$ to perform operation as a legitimate user. Further, timestamp $\mathcal{TSTP}_{\mathcal{UAV}}$ can be generated for access permissions in the framework. However, session based approach is applied to verify the unique identity of the devices $\mathcal{UAV}$. If all credential is matched then access permissions granted, else connections terminated immediately. Thus, this approach prevents from impersonation attack.

(2) Insider Attack: The attackers are privileged (can be insider) and can sniff all the credential like UAV identification $\mathcal{UAV}$, pseudo identity $\mathcal{PSID}_{\mathcal{UAV}}$, and timestamp $\mathcal{TSTP}_{\mathcal{UAV}}$. However, the access can only be permitted after session based verification of the entities. Thus, the approach does not allow access without permissions and prevents from insider attack.

(3) MITM and Replay attack: The attacker may get the details of the UAV from insecure channel and communications like $\mathcal{UAV}$ and timestamp $\mathcal{TSTP}_{\mathcal{UAV}}$ of registration. The attackers may send the details to the UAVs for making certain operations. However, the UAVs checks for the timestamp and verifies the session. However, it is difficult to compute all the credential at certain interval of time from id generation to session verification. Performing all the required evaluation at perfect time edge is difficult. Thus, the attacker cannot perform the MITM and replay attack.

5 PERFORMANCE ANALYSIS

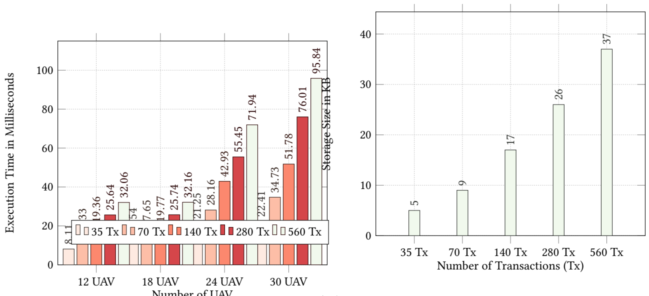
The proposed framework was implemented on a Tyrone PC with two 2.20GHz Intel CPUs and 128 GB of RAM. We have simulated the proposed framework on the Ethereum Rinkby blockchain platform using POA consensus algorithm. Ethereum Ropsten uses smart



(a) Tx Upload Time over IPFS

(b) Block Mining Time

Figure 2: Analysis of blockchain scheme



(a) Block Creation Time

(b) Transaction size in KB over IPFS

Figure 3: Analysis of blockchain scheme

contracts to specify the business logic without knowing the internal architecture of the blockchain system. The entity which is part of the consensus process is known as minters. The smart contracts are implemented in Solidity version 0.8.15. The wallet of each entity is created using metamask version 10.15.0. In proposed framework, we have considered block height, Transactions, gas used, gas limits, ECDSA signature, public key, current hash, previous hash are of sizes 32 bit, 1024 bits, 512 bit, 20 bytes, keccak-256 bit, and keccak-256 bit. The gas limit denotes the maximum amount of gas spent over a transactions. The higher gas limit means more computational work required to execute a transactions. The more gas amount is set to perform complex transactions in network. The computation of hash in proposed framework is evaluated using SHA-256.

5.1 Blockchain Result Analysis

The Fig.(3) shows the blockchain result analysis, where transactions upload time, block mining time, block creation time, and transaction off-chain storage size is computed. The Fig. (2a) shows transaction upload time for varying number of UAV and transactions. From the observed computation, the upload time is dependent on number of transactions. The Fig.(2b) and (3a) shows the block mining time and block creation time in the proposed framework. It can be seen that, the computation time in both the cases are dependent on number of UAV and number of transactions shared in the network. The Fig.(3b) shows the off-chain storage size of transactions which is computed in KB. It can be noticed that, the storage size in dependent on number of transactions stored in the off-chain layer.

6 CONCLUSION AND FUTURE WORK

In this article, we discussed the security aspect of secure UAVs communication and designed a novel secure authentication and key agreement framework. The proposed framework preserves four factor authentication of an entity i.e., unique identity, pseudo-identity, certificate, and timestamp. The framework supports session key establishment with mutual authentication, to make secure communication between entities. The data received by P2P cloud data center uses PoA consensus algorithm to construct, verify and write transactions into blockchain. The blockchain-enabled distributed cloud makes data "temperproof", "trustable" and rescues data from poisoning attack. Future work includes implementing proposed framework in realtime 5G-enabled UAV network.

REFERENCES

[1] Ahamed Aljuhani. 2021. Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access* 9 (2021), 42236–42264.

[2] Moayad Aloqaily, Ouns Bouachir, Azzedine Boukerche, and Ismaeel Al Ridhawi. 2021. Design guidelines for blockchain-assisted 5G-UAV networks. *IEEE network* 35, 1 (2021), 64–71.

[3] Basudeb Bera, Anusha Vangala, Ashok Kumar Das, Pascal Lorenz, and Muhammad Khurram Khan. 2022. Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment. *Computer Standards & Interfaces* 80 (2022), 103567.

[4] Ran Canetti and Hugo Krawczyk. 2002. Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 337–351.

[5] Ashok Kumar Das, Basudeb Bera, Sourav Saha, Neeraj Kumar, Ilsun You, and Han-Chieh Chao. 2021. AI-Envisioned Blockchain-Enabled Signature-Based Key Management Scheme for Industrial Cyber–Physical Systems. *IEEE Internet of Things Journal* 9, 9 (2021), 6374–6388.

[6] Danny Dolev and Andrew Yao. 1983. On the security of public key protocols. *IEEE Transactions on information theory* 29, 2 (1983), 198–208.

[7] Srinivas Jangirala, Ashok Kumar Das, and Athanasios V Vasilakos. 2019. Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment. *IEEE Transactions on Industrial Informatics* 16, 11 (2019), 7081–7093.

[8] Teng Li, Jianfeng Ma, Qingqi Pei, Chengyan Ma, Dawei Wei, and Cong Sun. 2019. Privacy-preserving verification and root-cause tracing towards UAV social networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[9] Ishan Mistry, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. 2020. Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges. *Mechanical systems and signal processing* 135 (2020), 106382.

[10] Syed Ahsan Raza Naqvi, Syed Ali Hassan, Haris Pervaiz, and Qiang Ni. 2018. Drone-aided communication as a key enabler for 5G and resilient public safety networks. *IEEE Communications Magazine* 56, 1 (2018), 36–42.

[11] Tarun Rana, Achyut Shankar, Mohd Kamran Sultan, Rizwan Patan, and Balamurugan Balusamy. 2019. An intelligent approach for UAV and drone privacy security using blockchain methodology. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 162–167.

[12] Anusha Vangala, Anil Kumar Sutrala, Ashok Kumar Das, and Minho Jo. 2021. Smart contract-based blockchain-envisioned authentication scheme for smart farming. *IEEE Internet of Things Journal* 8, 13 (2021), 10792–10806.

[13] Jian Wang, Yongxin Liu, Shuteng Niu, Houbing Song, Weipeng Jing, and Jiawei Yuan. 2021. Blockchain enabled verification for cellular-connected unmanned aircraft system networking. *Future Generation Computer Systems* 123 (2021), 233–244.

[14] Abbas Yazdinejad, Reza M Parizi, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2019. Blockchain-enabled authentication handover with efficient privacy protection in SDN-based 5G networks. *IEEE Transactions on Network Science and Engineering* 8, 2 (2019), 1120–1132.