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

Randhir Kumar, Ahamed Aljuhani, Prabhat Kumar, Abhinav Kumar, Antony Franklin and Alireza Jolfaei

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 the 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.

Index Terms—5G, Authentication and Key Agreement, Blockchain, Unmanned Aerial Vehicles.

I. 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) [1]; 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 [2]. 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 [3].

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 [4]. 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 [5]. 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 [6].

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 [7]. 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 [8], [9]. 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 [10]. 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 [11], [12]. 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 a smart contract facility makes up the cloud server. 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.

II. 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:

A. 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 (∪A), 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.

B. Threat Model

The "Dolev-Yao", often known as the DY model, is the first threat model that we employ in this paper [13]. According to this model, an adversary known as 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 (∪A) 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 [14]. In this instance, an adversary A has the ability to hijack a live session between two network users by stealing their secret credentials and the session key/state.

III. THE PROPOSED FRAMEWORK

A. Proposed Authentication and Key Agreement Module

1) Initialization Phase: This phase explores, how Trusted Authority (∪A) 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 TA i.e., $E_4(β, γ) S^2 = T^3 + αT + γ (mod \ W_0)$, where $W_0$ is a large prime value and $β, γ ∈ \mathbb{V}^* = \{1, 2, 3, ..., W_0\}$ are the two points i.e, infinity point and zero point ZO. Further, the TA chooses a base point $BP ∈ E_4(β, γ)$ of order $\mathbb{U}$ as bigger as $W_0$. Furthermore, $TA$ chooses a cryptographic hash function i.e.: $Hf(\cdot)$ using SHA-256. In addition, $TA$ chooses an identity $ID_{TA}$, and picks a private key $TAP_{PR} ∈ \mathbb{V}^*$ and evaluates a public key $TAP_{pub} = TAP_{PR} * BP$. Finally, the $TA$ preserve a private key $(TA_{PR})$ secret and disseminates public parameters{$(E_4(β, γ), BP, Hf(\cdot), TAP_{pub})$}.
2) Registration Phase: This phase describes a registration process of each entity and shares the communication parameters. (a) UAV Registration: The TA registers a UAV node \(\text{UAV} \), where \(\text{UAV} = \{1, 2, \ldots, n\}\).

Step-1: The TA chooses an unique identity \(ID_{\text{UAV}}\) for registration of UAV devices. Further, the TA evaluates a pseudo identity \(PSID_{\text{UAV}} = \text{HF}(ID_{\text{UAV}}) \) where \(PSID_{\text{UAV}}\) is a registration code of \(ID_{\text{UAV}}\) and generates a certificate \(CRT_{\text{UAV}} = \text{TPRP} + \text{HF}(PSID_{\text{UAV}}) \) and transmits the public key \(PB_{\text{UAV}}\) to the \(\text{UAV}\).

Step-2: A random number \(RN\) is chosen from \(\text{UAV} \), and evaluates a partial private key \(PPK_{\text{UAV}} = \text{HF}(TPRP) \) and generates a certificate \(CRT_{\text{UAV}} = TSTP + \text{HF}(PSID_{\text{UAV}})\) and transmits the public key \(PB_{\text{UAV}}\) to the \(\text{UAV}\).

(b) ES Registration: The TA registers a ES \(\text{ES}_1\), where \(\text{ES}_1 = \{1, 2, \ldots, n\}\).

Step-1: The ES chooses an unique identity \(ID_{\text{ES}}\) for registration of ES. Further, the ES evaluates a pseudo identity \(PSID_{\text{ES}_1} = \text{HF}(ID_{\text{ES}_1}) \) where \(PSID_{\text{ES}_1}\) is a registration code of \(ID_{\text{ES}_1}\) and generates a certificate \(CRT_{\text{ES}_1} = TSTP + \text{HF}(PSID_{\text{ES}_1})\) and transmits the public key \(PB_{\text{ES}_1}\) to the \(\text{ES}_1\).

Step-2: A random number \(RN\) is chosen from \(\text{ES}_1\), and evaluates a partial private key \(PPK_{\text{ES}_1} = \text{HF}(TPRP) \) and generates a certificate \(CRT_{\text{ES}_1} = TSTP + \text{HF}(PSID_{\text{ES}_1})\) and transmits the public key \(PB_{\text{ES}_1}\) to the \(\text{ES}_1\).

(c) CS Registration: The TA registers cloud servers \(\text{CS}_1\), where \(\text{CS}_1 = \{1, 2, \ldots, n\}\).

Step-1: The TA chooses an unique identity \(ID_{\text{CS}_1}\) for individual \(\text{CS}_1\) registration. Next, the TA evaluates pseudo identity \(PSID_{\text{CS}_1} = \text{HF}(ID_{\text{CS}_1}) \) where \(PSID_{\text{CS}_1}\) is a registration code of \(ID_{\text{CS}_1}\) and generates a certificate \(CRT_{\text{CS}_1} = TSTP + \text{HF}(PSID_{\text{CS}_1})\) and transmits the public key \(PB_{\text{CS}_1}\) to the \(\text{CS}_1\).

Step-2: A random number is chosen by TA i.e. \(RN\) from \(\text{CS}_1\), and evaluates a partial private key \(PPK_{\text{CS}_1}\) and generates a certificate \(CRT_{\text{CS}_1} = TSTP + \text{HF}(PSID_{\text{CS}_1})\) and transmits the public key \(PB_{\text{CS}_1}\) to the \(\text{CS}_1\).

3) Key Agreement and Authentication Phase: We have discussed various steps used in key agreement and authentication.

(i) UAV to UAV Authentication

Step-1: \(\text{UAV}_1\) picks an unique random value \(d_1\) and its current timestamp \(TSP_1\) and evaluates \(L_1 = h(PSID_{\text{UAV}_1}, PPK_{\text{UAV}_1}, d_1) \) where \(h\) is a cryptographic hash function. Further, \(\text{UAV}_1\) applies encryption as \(L_1 = E_{PB_{\text{UAV}_1}}(L_1)\). Furthermore, \(\text{UAV}_1\) evaluates the \(L_1 = \text{HF}(L_1, CRT_{\text{UAV}_1}, PSID_{\text{UAV}_1}, PPK_{\text{UAV}_1}, TSP_1)\) and sends the request message as \(M_1 = \{PSID_{\text{UAV}_1}, PPK_{\text{UAV}_1}, TSP_1, L_1\}\) and transmits to \(\text{UAV}_2\) through open channel.

Step-2: after successful receiving of message \(M_1\) by \(\text{UAV}_2\) timestamp gets validated i.e., \(TSP_2^* \) using \(TSP_1^* \) and \(TSP_1^* < \Delta T\). If it is valid \(\text{UAV}_2\) checks for certificates \(CRT_{\text{UAV}_1}\), \(B = PB_{\text{UAV}_1} + \text{HF}(PB_{\text{UAV}_1})\) if validated successfully, then \(\text{UAV}_2\) evaluates \(PSID_{\text{UAV}_1}\) with respect to \(PPK_{\text{UAV}_1}\) from the databases and evaluates \(L_1^* = \text{HF}(L_1, PSID_{\text{UAV}_1}, PPK_{\text{UAV}_1}, CRT_{\text{UAV}_1})\) to verify whether \(L_1^* = L_1\). If validated successfully, then \(\text{UAV}_2\) applies decryption \(L_2\) as \(L_2 = D_{PB_{\text{UAV}_1}}(L_1)\).

(ii) UAV to Edge Server (ES) Authentication

Step-1: \(\text{UAV}_1\) picks an unique random value \(d_2\) and its current timestamp \(TSP_2\) and evaluates \(L_2 = h(PSID_{\text{UAV}_2}, PPK_{\text{UAV}_2}, d_2) \) where \(h\) is a cryptographic hash function. Further, \(\text{UAV}_1\) applies encryption as \(L_2 = E_{PB_{\text{UAV}_2}}(L_2)\). Furthermore, \(\text{UAV}_1\) evaluates the \(L_2 = \text{HF}(L_2, CRT_{\text{UAV}_2}, PSID_{\text{UAV}_2}, PPK_{\text{UAV}_2}, TSP_2)\) and sends the request message as \(M_2 = \{PSID_{\text{UAV}_2}, PPK_{\text{UAV}_2}, TSP_2, L_2\}\) and transmits to \(\text{ES}\) through open channel.

Step-2: once the reply message is received by \(\text{ES}\) \(M_2\) checks for stamps \(TSP_2^* - TSP_3^* < \Delta T\). If validated successfully, the \(\text{ES}\) checks for certificates \(CRT_{\text{UAV}_2}\), \(B = PB_{\text{UAV}_2} + \text{HF}(PB_{\text{UAV}_2})\) if matched successfully, then \(\text{ES}\) proceeds for update \(PPK_{\text{UAV}_2}\) and \(PPK_{\text{UAV}_2}^*\) into the database.

Step-3: Once acknowledgement receipt is received by \(\text{UAV}_2\), \(\text{M}_2\) checks for stamps \(TSP_2^* - TSP_3^* < \Delta T\). Further, \(\text{UAV}_2\) checks \(SECS_{\text{UAV}_2} = \text{HF}(SECS_{\text{UAV}_1}, TSP_3)\) and computes \(SECS_{\text{UAV}_2} = SECS_{\text{UAV}_2}^* = \text{HF}(SECS_{\text{UAV}_2}, TSP_3)\) and goes for update of \(PPK_{\text{UAV}_2}\) and \(PPK_{\text{UAV}_2}^*\) into the database. Furthermore, \(\text{UAV}_2\) generates acknowledgment receipt \(M_2 = \{SECS_{\text{UAV}_2}, TSP_3\}\) and transmits to \(\text{UAV}_1\) through open channel.

Step-4: if no acknowledgement receipt is received by \(\text{UAV}_2\), \(\text{M}_2\) checks for stamps \(TSP_2^* - TSP_3^* < \Delta T\). Further, \(\text{UAV}_2\) checks \(SECS_{\text{UAV}_2} = \text{HF}(SECS_{\text{UAV}_1}, TSP_3)\) and computes \(SECS_{\text{UAV}_2} = SECS_{\text{UAV}_2}^* = \text{HF}(SECS_{\text{UAV}_2}, TSP_3)\) and goes for update of \(PPK_{\text{UAV}_2}\) and \(PPK_{\text{UAV}_2}^*\) into the database.
Step 4: once reply messages is successfully received (M₃) from ES then timestamp gets validated through open channel.

Step 5: once acknowledgement received successfully, M₄ then timestamp gets validated i.e., Tₙ₄ by CS | Tₙ₄ - Tₙ₃ < ΔT. Further, ES verification is applied through verification i.e., ESₜ₄(=ESₜ₃) to UAV. Further, ES goes for updation of PPPₜ₄ and PPPₜ₃ into the database securely.

B. Proposed Permissioned Blockchain-based Transaction Writing Module

This phase describes the block verification and addition. The authorized UAVs are responsible to create the transactions in the network. Next, the transactions gets verified by the authorized ES denoted as miner. Further, block gets created and added into the blockchain. The block consists of two parameters namely Vᵢ ← (Wᵢ, Xᵢ), where Wᵢ denotes local blockchain ledger of peers where Xᵢ denotes block pointer. The algorithm consists of four different functions i.e. 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 CTₛₜ₃, 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.,
Algorithm 1 Proof-of-Authority (Round-based Aura Algorithm) for Block Verification and Addition

1: State: $\mathcal{E}_S \in \mathcal{B}_{\mathcal{N}_{\mathcal{S}}}$ Set of miners,
2: $\mathcal{W}_i = \mathcal{W}_i \cdot \mathcal{W}_i \cdot \mathcal{W}_i \cdot \mathcal{W}_i$ local blockchain of node $\mathcal{X}_i$ is a DAG of block $\mathcal{W}_i$ and point $\mathcal{X}_i$
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 to validate and added /*miner keep proposing a block */
8: function PROPOSE():
9:     while True do
10:         sleep $\leftarrow$ CTS / duration, CTS $\rightarrow$ clock time
11:     if $k \in \mathcal{E}_S \land$ step mod $|\mathcal{E}_S|$ == $k$ then
12:         $b.parent \leftarrow lb(b)$, $lb \rightarrow$ last block
13:         $b.ES \leftarrow x_i$
14:         $b.step \leftarrow$ step
15:         $\mathcal{W}_i \leftarrow (\mathcal{W}_i \cup b, x_i \cup b.parent )$
16:     /* disseminate calls DELIVER() function internally */
17:         disseminate ($\mathcal{W}_i$)
18:         sleep(duration)
19:     end if
20: end while:
21: end function
22: /* returns a score for correct height of blockchain */
23: function SCORE($\mathcal{W}_i$, $x_i$)
24:     return UNIT256-MAX * height($\mathcal{W}_i$, $x_i$)
25: end function
26: /* Deliver function computes score with correct height of blockchain */
27: function DELIVER($\mathcal{W}_i$, $x_i$)
28:     if $\text{Score}(\mathcal{W}_i, x_i) > \text{Score}(\mathcal{W}_i, x_i)$
29:         $\text{Score}(\mathcal{W}_i, x_i) \leftarrow \text{Score}(\mathcal{W}_i, x_i)$
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($b$)
34:     $p \leftarrow ID_S$
35:     round1 $\leftarrow$ (b.step, b.step + p)
36:     round2 $\leftarrow$ (b.step + p, b.step + 2*p)
37:     majority1 $\leftarrow$ { $b' : b', \text{step} \in \text{round1} $}
38:     majority2 $\leftarrow$ { $b' : b', \text{step} \in \text{round2} $}
39:     return (majority1 $\cap$ majority2)
40: end function

round1 and round2 for block $b$, in each round block $b$ is mined (verified) by majority of miners ($\mathcal{E}_S$). Next, block $b$ gets added into the blockchain network after successful mining (verification). The block verification and block Addition is detailed in Algorithm 1.

IV. 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 $\mathcal{UA}_V$, pseudo identity $PSID_{\mathcal{UA}_V}$, and partial private key $PPR_{\mathcal{UA}_V}$ to perform operation as a legitimate user. Further, timestamp $TSTP_{\mathcal{UA}_V}$ can be generated for access permissions in the framework. However, session based approach is applied to verify the unique identity of the devices $\mathcal{UA}_V$. 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 $\mathcal{UA}_V$, pseudo identity $PSID_{\mathcal{UA}_V}$, and timestamp $TSTP_{\mathcal{UA}_V}$. 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 details of the UAV from insecure channel and communications like $\mathcal{UA}_V$ and timestamp $TSTP_{\mathcal{UA}_V}$ 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.

V. 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 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 miners. 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.

A. 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.
VI. 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


