01 Jan 2023

Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks

Anusha Vangala

Ashok Kumar Das

Ankush Mitra

Sajal K. Das
Missouri University of Science and Technology, sdas@mst.edu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/comsci_facwork/1263

Recommended Citation

A. Vangala et al., "Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 904 - 919, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/TIFS.2022.3231121

Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks

Anusha Vangala, Ashok Kumar Das, *Senior Member, IEEE*, Ankush Mitra, Sajal K. Das, *Fellow, IEEE*, and Youngho Park, *Member, IEEE*

***Abstract*— Precision farming has a positive potential in the agricultural industry regarding water conservation, increased productivity, better development of rural areas, and increased income. Blockchain technology is a better alternative for storing and sharing farm data as it is reliable, transparent, immutable, and decentralized. Remote monitoring of an agricultural field requires security systems to ensure that any sensitive information is exchanged only among authenticated entities in the network. To this end, we design an efficient blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural Internet of Things (IoT) networks called *AgroMobiBlock*. The limited existing work on authentication in agricultural networks shows passive usage of blockchains with very high costs. *AgroMobiBlock* proposes a novel idea using the elliptic curve operations on an active hybrid blockchain over mobile farming vehicles with low computation and communication costs. Formal and informal security analysis along with the formal security verification using the Automated Validation of Internet Security Protocols and Applications (AVISPA) software tool have shown the robustness of *AgroMobiBlock* against man-in-the-middle, impersonation, replay, physical capture, and ephemeral secret leakage attacks among other potential attacks. The blockchain-based simulation on large-scale nodes shows the computational time for an increase in the network and block sizes. Moreover, the real-time testbed experiments have been performed to show the practical usefulness of the proposed scheme.**

***Index Terms*— Intelligent precision agriculture, Internet of Things (IoT), mobile vehicles, blockchain, authentication and key agreement, security, simulation.**

I. Introduction

THE sustainability of humankind is highly dependent on the prosperity of the food and agriculture industry. According to World Population Prospects, 2019 [1], the population will rise globally to 8.5 million by 2030, 9.7 billion by 2050, and 10.9 billion by 2100. The growth of the population increases the demands on the food industry. The pandemics, disasters, and natural/human-induced calamities directly affect the amount of food produce available to the general population. Along with the quantity of production, the quality of produce plays a vital role in the general health of the public. Ramakumar [2] studied the effect of the Coronavirus disease (COVID-19) pandemic on the agriculture sector worldwide, specifically in India. This study shows that the lack of labor for fieldwork during the lock-down imposed in 2020 resulted in the reduction of total arrival of crops into the agricultural market by 55.6% for wheat, 26.9% for gram, 25.9% for mango, 40.2% for barley of the respective crop amount compared to the produce in 2019.

A remote monitoring system uses sensing and automation technologies to supervise an IoT-based agriculture field. It has a great relevance during the work-from-home culture of the pandemic. Remote monitoring systems transmit data related to soil condition, crops, effect of used chemicals on the crops, quantity of yield, quality of yield, and time of yield. These data affect the price of the yield received by the stakeholders. However, remote monitoring systems are highly vulnerable to several cyberattacks from unauthorized parties leading to loss of confidentiality, integrity, and data availability.

The farming data from a country's numerous states/provinces creates a huge volume of data that becomes a national or global food security concern. Such data is significant in an economic, natural, or human-induced crisis, such as COVID-19 pandemic [3], [4], and political wars [5], [6]. These situations affect a country's economy by forcing its government to increase or decrease imports and exports as the need arises. When the national farming data is available to unauthorized entities, such as hostile/rival governments or extremist/radical groups, they are prone to bio-wars

Manuscript received 10 May 2022; revised 26 July 2022 and 30 November 2022; accepted 6 December 2022. Date of publication 20 December 2022; date of current version 30 December 2022. This work was supported in part by the National Mission on Interdisciplinary Cyber-Physical Systems (NM-ICPS) for the Technology Innovation Hub (FINTECH) at the Indian Institute of Technology (IIT) Bhilai, Department of Science and Technology, Government of India; and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605. The work of Sajal K. Das was supported in part by the NSF Grants Smart Integrated Farm Network for Rural Agricultural Communities (SIRAC) under Award 1952045, in part by the Towards a Unified Robust and Secure Data Driven Approach for Attack Detection in Smart Living (TAURUS) under Grant 2030624, and in part by the Robust Federated Learning for Internet of Things (FLINT) under Grant 2008878. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dusit Niyato. *(Corresponding authors: Ashok Kumar Das; Youngho Park.)*

Anusha Vangala, Ashok Kumar Das, and Ankush Mitra are with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India (e-mail: anusha.vangala@research.iiit.ac.in; iitkgp.akdas@gmail.com; ankush.mitra@alumni.iiit.ac.in).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Youngho Park is with the School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea (e-mail: parkyh@knu.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2022.3231121, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2022.3231121

that deliberately target the country's fundamental national resource. Hence, there must be proper security protocols to prevent untrusted parties from accessing data circulating in a remote farm monitoring system. Widespread adoption of remote monitoring systems is possible only when they guarantee security against attacks. Authentication schemes for remotely monitored environments ensure that sensitive information is exchanged only between authenticated parties. A secure remote monitoring system needs to protect the data in transmission and storage from misuse. Such protection of data increases trust in the data. Thus, secure, trusted data promotes informed decision-making to ward off social issues like food crises, low production, and bio-war.

The transfer of sensor data from a farm to a central storage and processing unit is a core aspect of a smart farming system. Authentication ensures that data is collected only from reliable sources and delivered only to trustworthy entities. This high-level problem statement is fractionalized into the following low-level challenges: a) Sensors must verify themselves securely to an entity requesting their data before establishing an encryption/decryption key in every session, b) Any entity communicating with the sensor must verify its authenticity in every session, and c) Any entity that holds or transmits the sensor data either temporarily or permanently in either encrypted on unencrypted format must authenticate itself before obtaining access to the data. A secret key to encrypt the data transmission must be agreed upon in every session. An attacker is capable of man-in-the-middle (MiTM), impersonation, privileged insider, replay, ephemeral secret leakage (ESL), denial of service (DoS), and physical capture attacks, which are considered to test the designed scheme in this work.

Blockchain technology is essentially a ledger that stores information significant to an application. The necessity and challenges of blockchain in precision agriculture along with the relevant use cases in extensive detail were provided in [7]. Specifically, in precision agriculture, blockchain can be used to record data, verify properties, track and monitor movables, link products to tags or codes, and share information about agri-products. Moreover, a tool was suggested in [8] to evaluate the need for blockchain in smart farming and choose the type of blockchain based on several criteria.

An active blockchain stores and retrieves both data and secret credentials simultaneously without the deployment of smart contracts, while the registration and authentication are in progress. The transaction contents in a block from an active blockchain have secret credentials that are required to execute the cryptographic operations in the authentication. Without these credentials, the authentication process would be unable to execute all its steps, and it will lead to a failed authentication. An inactive or passive blockchain is accessed independently of authentication. It does not affect the outcome of authentication between the entities. The full potential of a passive blockchain is not harnessed as its use is limited to one phase and stores either credentials or data but not both.

Most existing authentication schemes in smart agriculture [9], [10], [11], [12], [13], [14], [15] do not use blockchain technology. The scheme in [16] uses the blockchain technology, but it is inefficient due to its high cost. The schemes proposed in [17] and [18] are relatively efficient. However, all these use the blockchain passively only to store the data from the sensing equipment in the IoT networks. Thus, the blockchain does not play any role during the authentication process. Moreover, no existing works have attempted to use the active blockchain alongside elliptic curve operations. None of the schemes have used agricultural vehicles in their model. Thus, no current schemes have considered using elliptic curve cryptography (ECC) together with mobile vehicles and hybrid blockchain to achieve mutual authentication in smart agriculture.

A smart agriculture environment uses several vehicles such as tractors, harvesters, farm trucks, bale handlers, balers, crop sprayers, front-end loaders, lawn mowers, rollers, cultivators, harrows, subsoilers, seed drills, land imprinter, stone picker, manure spreader/honey wagon, tree shaker, swather, and several other machines. These machines may be manually driven or operated autonomously. The idea for this work stems from the fact that vehicles intrinsic to the farming process should be put to good use for secure data transmission in a smart farm. The role of blockchain in the current work is to store secret credentials for authentication along with sensor data. Only part of the data collected in smart agriculture needs to be encrypted while the rest of the data resides unencrypted. Therefore, hybrid blockchain is most appropriate for smart agriculture [17]. Each block in the proposed hybrid blockchain consists of partly encrypted transactions to store credentials or fully encrypted transactions to store sensor data. It is well-known that ECC supports design of lightweight schemes. Hence, ECC involving hybrid blockchain over fog servers with mobile vehicles as data collectors has a valuable potential to achieve mutual authentication in smart agriculture.

The novel contribution of the blockchain part in the proposed scheme lies in the increased potential use of blockchain in multiple phases to store data and credentials together. Access to the blockchain is inevitable during authentication as it holds critical credentials from registration. It has the advantage of avoiding the privileged-insider attack, which is an essential attack in any authenticated key agreement scheme. In addition, the blockchain is also used for storing sensor data rather than storing it in semi-trusted cloud servers. If we keep the data in semi-trusted cloud servers, there are possibilities of data poisoning attacks that are very crucial concerns, and they may cause a significant factor for the businesses and organizations for both financial terms as well as damaging their reputations when the Big data analytics are performed on the analyzed data which becomes corrupted [19].

The novel contributions of this paper are summarized below.

- The blockchain is leveraged to its full potential by using it in multiple phases. Specifically, we propose a new blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks, called $AgroMobiBlock$, which makes use of an active consortium blockchain. To the best of our knowledge, this is the first attempt to use an active blockchain alongside elliptic curve operations in smart farming.

- The proposed scheme leverages vehicular farming systems during the authentication process, which has not been explored in the existing farming applications.
- A real-time implementation using testbed setup gives the step-wise execution time of each phase of $AgroMobiBlock$. In addition, the blockchain simulation observes that the consensus time has a significant increase with the number of nodes and a small increase with the number of transactions. Increasing the number of transactions per block increases the throughput, but it reduces the service time of the blockchain as well.

The rest of the paper is organized as follows. Section II reviews the related work while the system models are introduced in Section III. The proposed scheme ($AgroMobiBlock$) is presented in Section IV. The security analysis related to $AgroMobiBlock$ is provided in Section V, while the formal security verification is given in Section VI. Section VII provides a comparative analysis of $AgroMobiBlock$ with other relevant existing schemes. In Section VIII, we supply the real-time implementation of the proposed model with testbed experiments, and Section IX provides a blockchain implementation of $AgroMobiBlock$. Finally, Section X offers concluding remarks.

II. Related Work

We provide a review on authentication schemes in smart agriculture and other relevant schemes in smart networks.

A. Review of Existing Authentication Schemes

1) Authentication in IoT-Based Smart Agriculture Networks: Except for the work in [16], no other literature reviewed here relies on blockchain technology. The earliest work in [9] proposed a user authentication scheme for monitoring an agricultural wireless sensor network using hash functions and symmetric cryptography. It is resistant to replay attacks, deployment of malicious devices, and device capture attacks. This work is enhanced in [10] to add untraceability and user anonymity by allowing the security parameters to be dynamic instead of static. In addition, it enables perfect forward secrecy and adds resistance to distributed Denial-of-Service (DoS), privileged insider, ephemeral secret leakage (ESL), and user impersonation attacks.

In [11], the authors proposed an authentication scheme for a user to access devices internal to a glasshouse via external devices using advertisement. The internal devices can authenticate both external devices and users. This scheme can only resist brute force and dictionary attacks, which are trivial in smart farming, and does not use blockchain technology. The authors in [12] use the "Constrained Application Protocol (CoAP)" [20] to develop an authentication framework while the authors in [13] uses the "project open software platform for service innovation in a value added network for agriculture (ODiL)" platform on "Open Authorization (OAuth)" framework with "radio-frequency identification (RFID)" to poll credentials.

More recent works that are conceptually close to our proposed system are the ECC-based scheme [15] and the blockchain-based scheme [16]. However, the former scheme lacks anonymity, untraceability, and dynamic node addition; while the latter is vulnerable to ESL and offline guessing attacks, and is also costly due to the usage of bilinear pairings.

In the survey [21], the use of blockchain for providing authentication in agricultural IoT networks has been explored. In another survey [22], the existing works are studied that fulfill various security objectives in smart agriculture, especially authentication, without the use of blockchain. Both surveys conclude that existing protocols are insufficient for achieving the required security requirements. The work in [17] is based on an authentication scheme for smart agriculture using a hybrid blockchain with smart contracts. Another work in [18] uses the private blockchain in an agricultural environment in conjunction with unmanned aerial vehicles (UAVs). However, it can store only classified data specific to a part of the stakeholder community. None of these works use the concept of an active blockchain or the use of mobile agricultural vehicles.

2) Authentication in Generalized IoT-Based Smart Networks: We review some recent authentication and key agreement schemes from diverse use cases that may be used in smart farming. First, we discuss those schemes that do not use blockchain. The RSA-based scheme [23] is costly, and it does not provide anonymity or untraceability, and is vulnerable to various attacks (e.g., DoS, offline guessing, ESL, and impersonation attacks) with no support for blockchain technology. The Rabin cryptosystem-based scheme [24] provides anonymity and untraceability, but it is not resistant to such attacks as stolen smart card, stolen mobile device, privileged insider, ESL, and offline password guessing attacks, with high communication cost. The scheme in [25] uses one-way hash chains, but it does not support anonymity, untraceability, and addition of nodes dynamically. It is also vulnerable to ESL, offline guessing, stolen smart card or mobile device, impersonation, and privileged insider attacks.

We now discuss some blockchain-based schemes in diverse applications. The ECC-based method in [26] has an average computational performance and low communication cost, but it lacks anonymity, untraceability and dynamic node addition, and it is also exposed to stolen smart card, stolen mobile device, DoS, privileged insider, and ESL attacks. The scheme proposed in [27] applies the bilinear pairings that has a hefty computation cost with no anonymity, untraceability, and dynamic node addition. This scheme is also exposed to various attacks like smart card, stolen mobile device, ESL, DoS, and offline guessing attacks. Another ECC-based approach [28] has reasonable computational and communication costs. However, it is vulnerable to stolen mobile device, ESL, privileged insider, DoS, and offline guessing attacks, and it does not support any dynamic node addition. The ECC-based schemes [29], [30] for healthcare domains are vulnerable to ESL, privileged insider, DoS, and offline guessing attacks. However, the scheme in [30] supports anonymity and untraceability features.

3) Dynamic Identity-Based Authentication Schemes: Among recent blockchain-based dynamic identity (DID) schemes, the one in [31] is vulnerable to impersonation, ESL, privileged insider, MiTM, DoS, and offline guessing attacks. The authors in [32] improved the scheme in [33] to prevent vulnerabilities to masquerade, password guessing, and stolen

smart card attacks. However, it is also vulnerable to privileged insider and DoS attacks, and it does not support dynamic node addition. In [34], the authors proposed an authentication model that assesses the risk of every user based on its usage and this controls access according to its risk behavior. The authors in [35] proved that the scheme in [36] does not achieve secure authentication and is vulnerable to ESL attacks.

B. Analysis of Studied Schemes

We analyze the usage of blockchain in the blockchain-based authentication schemes discussed above. The blockchain in [16] stores the packets received from the agricultural equipment. In [26], the authentication results are stored into the blockchain at the end of the authentication process. The authors in [17], [18], and [27] store the data from IoT nodes on the blockchain, whereas the scheme in [29] stores all the messages exchanged during the authentication over blockchain. Even though the blockchain in [30] is accessed during authentication (since it uses the blockchain as an alternative form of "Public Key Infrastructure (PKI)" instead of storing critical secrets for the authentication process), it does not conform to the concept of active blockchain. It stores only public keys of all entities, including the end users on the blockchain, via a smart contract. In addition, innumerable end users could make their blockchain large and unmanageable. Thus, these schemes use passive blockchain, which does not play any role during the authentication process. The current research aims to explore blockchain's usage in storing and retrieving crucial secrets while the authentication is in progress.

Table I provides a comparative analysis on various "security and functionality features" among the proposed scheme ($AgroMobiBlock$) and other existing schemes in [9], [10], [11], [15], [16], [23], [24], [25], [26], and [27]. The analysis reveals the need for the proposed $AgroMobiBlock$ as it achieves more security and functionality features compared to the existing authentication schemes. Comparing the proposed scheme with the existing ones, it is observed that $AgroMobiBlock$ has the ability to resist ESL attack while simultaneously achieving anonymity and untraceability properties for the IoT smart devices, mobile vehicles, and fog servers. In addition, the active blockchain presented in this work is not supported by any other relevant schemes without the need for smart contracts.

TABLE I

COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

Scheme	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
Authentication in Smart Agriculture without Blockchain														
Ali *et al.* [9]	×	×	✓	×	×	×	×	✓	✓	✓	✓	×	✓	NA
Chen *et al.* [10]	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓	✓	✓	NA
Chae and Cho [11]	×	×	×	×	NA	×	×	×	×	✓	✓	×	×	NA
Rangwani *et al.* [15]	×	×	×	✓	✓	×	✓	✓	✓	✓	✓	×	✓	NA
Authentication in Smart Agriculture with Blockchain														
Wu and Tsai [16]	✓	✓	×	×	NA	×	✓	✓	✓	✓	✓	✓	×	×
Vangala *et al.* [17]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Bera *et al.* [18]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Authentication in Diverse Applications without Blockchain														
Tian *et al.* [23]	×	×	✓	×	✓	×	✓	✓	✓	✓	✓	×	×	NA
Shuai *et al.* [24]	✓	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	×	NA
Panda *et al.* [25]	×	×	×	✓	NA	×	✓	✓	✓	✓	✓	✓	✓	NA
Authentication in Diverse Applications with Blockchain														
Eddine *et al.* [26]	×	×	×	✓	×	×	×	✓	✓	✓	✓	×	✓	×
Fan *et al.* [27]	×	×	×	✓	×	×	✓	✓	✓	✓	✓	×	×	×
Tomar & Tripathi [28]	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	×	×	×
Itoo *et al.* [29]	×	×	NA	✓	NA	×	×	✓	✓	✓	✓	×	×	×
Jia *et al.* [30]	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	×	×	×
DID Authentication														
Andola *et al.* [32]	✓	✓	NA	✓	×	×	×	×	×	✓	×	×	✓	NA
Liu *et al.* [34]	×	×	NA	×	×	×	×	✓	×	✓	✓	×	×	NA
Mishra *et al.* [31]	×	×	NA	×	×	×	×	✓	×	✓	✓	×	×	NA
Gupta *et al.* [36]	×	×	×	×	×	×	×	✓	×	×	×	×	×	NA
AgroMobiBlock	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: F_1 : "anonymity", F_2 : "untraceability", F_3 : "dynamic node addition", F_4 : 'device/user impersonation attacks", F_5 : "stolen mobile device attacks", F_6 : "ESL attacks", F_7 : "privileged insider attacks", F_8 : "replay attacks", F_9 : "MiTM attacks", F_{10} : "mutual authentication", F_{11} : "unauthorized login detection", F_{12} : "DoS attacks", F_{13} : "offline guessing attacks", F_{14} : "active blockchain (THIS WORK)". ✓: "Supports feature/resists attacks"; ×: "No support for feature/vulnerable to attacks"; NA: "Not applicable"

III. SYSTEM MODELS

A. Network Model

The proposed scheme is designed for a network model which is presented in Fig. 1. This architecture applies to many agricultural fields where each field is divided into disjoint regions. Hundreds of sensor nodes SN are scattered across the field to collect environmental readings. Various mobile vehicles MV are used for farming activities in agricultural field work as mobile sinks to collect data from the sensors in that field. The collected data is then sent to the fog computing layer. Each agricultural region is assigned to a fog server FS. The fog servers connected to the regions in a single agricultural field form a fog system to maintain a decentralized blockchain that can store and process data required during the authentication process, along with running a consensus algorithm. The proposed scheme uses the blockchain actively during the authentication process for managing critical parameters, along with passive data storage from the sensors after the authentication is completed.

The considered blockchain is a consortium blockchain that can have two types of blocks: a) AuthCred block, which stores the credentials needed during the authentication process, and b) SensorData block, which stores the sensor data received after encryption with the key established during the key management process. There are two types of AuthCred blocks, one is for mobile vehicles and the other is for the fog servers. For the sensor data to be securely stored in the blockchain, the first authentication is required between the sensors and the mobile vehicle. The second authentication is required between the mobile vehicle and the fog server, which takes credentials from the blockchain to verify the requesting vehicle. The sensor data is then forwarded from the sensor to the fog server via the mobile vehicle and stored on the blockchain. When the Big Data analytics center (BDAC) requires sensor data, it needs to send a request to the appropriate fog server, which retrieves the requested sensor data from the blockchain, encrypts it with the public key of the cloud storage inside BDAC, and sends it as a response to BDAC.

B. Threat Model

For a systematic analysis of the required defenses for the proposed authentication scheme, the standard "Dolev-Yao (DY) threat model" [37] and the current *de facto* "Canetti and Krawczyk's model (CK-adversary model)" [38] have been considered as the most appropriate threat models. An adversary $\mathcal{A}$ under the DY threat model can perform the following actions on the network model:

- All communication in public channels among the smart devices, mobile vehicles, fog servers, and cloud servers is accessible to $\mathcal{A}$ and allows it to seize, remove, modify and re-transmit existing messages or circulate counterfeit messages.

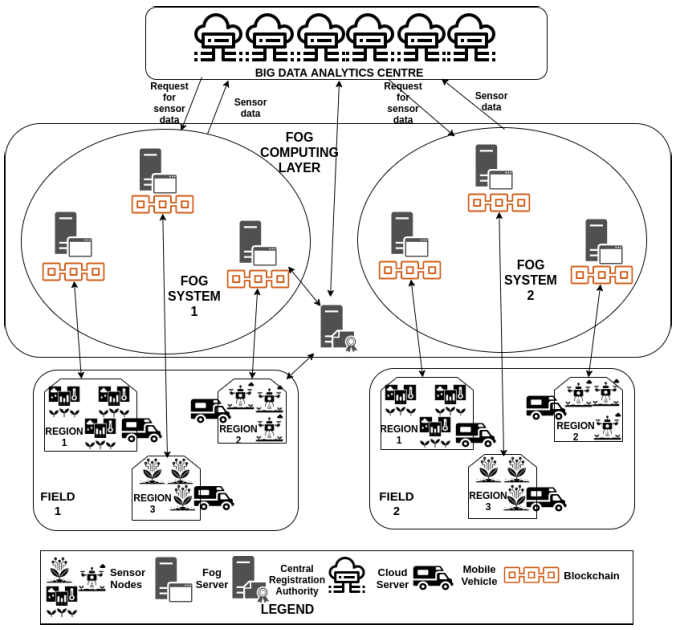


Fig. 1. Blockchain-based mobile vehicles-assisted precision agricultural IoT network.

TABLE II
NOTATIONS AND THEIR DESCRIPTIONS

Notation	Significance
$E_q(\kappa, \mu)$	A non-singular elliptic curve of the form: $y^2 = x^3 + \kappa x + \mu \pmod q$ over Galois field $GF(q)$
G	A base point in $E_q(\kappa, \mu)$ of order is n_G as big as q
$x \cdot G$	Elliptic curve point multiplication: $x \cdot G = G + G + \cdots + G\,(x\,times)$
$A + B$	Elliptic curve point addition; $A, B \in E_q(\kappa, \mu)$
TRA	Trusted Registration Authority
SN	IoT smart device
MV	Mobile vehicle
FS	Fog server
$BDAC$	Big data analytics center
RTS_X	Registration timestamp issued by the TRA to entity X
ID_M, TID_M, RID_M	MV's real identity, temporary identity, and pseudo-identity, respectively
ID_F, TID_F, RID_F	FS's real identity, temporary identity and pseudo-identity, respectively
ID_{SND}, TID_{SND}, RID_{SND}	SN's real identity, temporary identity and pseudo-identity, respectively
pr_{TRA}, Pub_{TRA}	Private and public key of TRA, respectively
pr_M, Pub_M	Private key and public key of MV, respectively
pr_F, Pub_F	Private key and public key of FS, respectively
pr_S, Pub_S	Private key and public key of SN, respectively
K_{MV_i,FS_j}	Association key between MV_i and FS_j
m, j_m, u_m	MV's random secrets
f, v_f	FS's random secrets
s, i_s	SN's random secrets
$\|\|$	Concatenation operation
TS_X	Current timestamp produced by an entity X
$*$	Integer multiplication operation
$\oplus$	Exclusive OR operation
ΔT	Maximum allowed delay in transmission for a particular message
$H(\cdot)$	"Collision-resistant cryptographic one-way hash function"

• $\mathcal{A}$ may impersonate a smart device, mobile vehicle, or fog server, and carry out tasks on their behalf.

• $\mathcal{A}$ may initialize multiple executions of the protocol simultaneously. Smart devices, mobile vehicles, fog servers, and cloud servers may take part in any number of such concurrent executions at the same time.

• The smart devices, mobile vehicles, fog servers, and cloud servers are honest and stateless, whereas $\mathcal{A}$ is stateful.

An adversary $\mathcal{A}$ under the CK-adversary threat model can perform the following actions on the network model:

• $\mathcal{A}$ enjoys all the capabilities as in the DY threat model.

• $\mathcal{A}$ can extract secret credentials by hijacking session states during communication among the communicating smart devices, mobile vehicles, fog servers, and cloud servers.

In addition, we assume that $\mathcal{A}$ can physically capture some smart devices as well as mobile vehicles to extract secret credentials from their memory using the power analysis attacks [39] and timing attacks [40]. The fog servers and cloud servers are assumed to be under a physical locking system as suggested in [41] and [42]. Thus, it is assumed that $\mathcal{A}$ cannot launch stolen verifier attacks on fog servers and cloud servers as all secret credentials stored on these servers are placed in their secure databases.

IV. THE PROPOSED SCHEME

In this section, we describe a new blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks, called $AgroMobiBlock$. Various notations that are used in this phase are provided in Table II with their descriptions.

A. High-Level Protocol Overview

The proposed scheme begins with a one-time registration of the involved entities of IoT smart devices, mobile vehicles, and fog servers. The registration of the mobile vehicle and fog server creates an AuthCred block, which stores the authentication credentials to be used during the authentication phase between a mobile vehicle and a fog server ($MVFS$). Part of each transaction in such a block is encrypted. There are two types of AuthCred blocks: 1) one is for storing private parameters for a fog server and 2) other is for storing private parameters for a mobile vehicle.

The first phase of authentication between an IoT smart device and a mobile vehicle ($SNMV$) is required between a sensor node and a mobile vehicle culminating in a session key agreement where the session key consists of a private hash from the sensor node and a private hash from the mobile vehicle. The second phase of authentication between a mobile vehicle and a fog server ($MVFS$) retrieves the long-term secret credentials from the blockchain to verify the requesting vehicle. A session key is then established, which consists of a long-term secret from a mobile vehicle encrypted using an association key, a private hash from the mobile vehicle, a private hash from the fog server, and a long-term secret from the fog server encrypted using the association key, and the Diffie-Hellman type key.

The sensor data is forwarded from the sensors to the fog server via the mobile vehicle using the established session keys in the $SNMV$ and $MVFS$ phases. The fog server creates a transaction out of the received sensor data and a SensorData block, out of a collection of such transactions. This block is then added to the single hybrid blockchain. To achieve the requirements that Basin et al. [43] proposed for entity authentication, each message includes the temporary and pseudo-identity of the sender, and the trusted registration authority (TRA) uniquely carries out the role of trusted authority.

The blockchain is truly hybrid in the sense that it consists of two types of blocks: 1) AuthCred block, which contain the registration credentials needed during the authentication

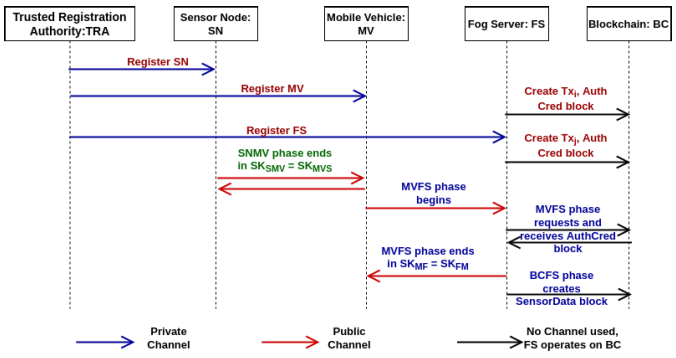


Fig. 2. Overview of the proposed $AgroMobiBlock$.

process, and 2) SensorData block, which contains the sensor data received securely during the authenticated key management process. Each transaction in the AuthCred block consists of partly encrypted and partly unencrypted contents. Critical content is first encrypted, combined with unencrypted content, and a transaction is created from both the unencrypted and encrypted contents. The transactions in the SensorData block consist of only unencrypted content inside. Encryption is applied to the entire transaction, and a collection of such encrypted transactions are made into blocks. Two different types of blocks are stored in a single hybrid blockchain, which reduces the overhead of using separate blockchains for storing authentication credentials and sensor data. Two different types of blocks for the separate storage of authentication credentials and sensor data help in keeping a clear distinction of access control between the two. Any access to sensor data will not reveal the authentication credentials used in any session and vice-versa. Hence, different levels of access control may be applied to them in the future.

Private channels are established directly between the TRA and the entities (SN, MV, and FS). There are no private channels among the entities SN, MV, and FS. These private channels are used during the registration phase as described in Section IV-C to send and receive private identities, private random secrets, and private session key parameters that should be accessible only to TRA and the registering entity. Public channels are established separately between the entities SN and MV, and the entities MV and FS. The messages communicated during the authentication process in Section IV-D and secure data aggregation phase in Section IV-E use these public channels. A symbolic representation of the proposed $AgriMobiBlock$ is shown in Fig. 2.

The proposed authentication and key agreement protocol ($AgroMobiBlock$) consists of the following phases as described below in detail.

B. System Initialization Phase

This phase involves the following steps:

Step S_1*:* The Trusted Registration Authority (TRA) selects a non-singular elliptic curve $E_q(\kappa, \mu) : y^2 = x^3 + \kappa x + \mu \pmod q$ over the Galois field $GF(q)$, with a "point at infinity (zero point)" $\mathcal{O}$, constants $\kappa, \mu \in Z_q = \{0, 1, 2, \cdots, q-1\}$ such that $4\kappa^3 + 27\mu^2 \neq 0 \pmod q$ is satisfied. The TRA picks a base point $G \in E_q(\kappa, \mu)$ whose order n_G as large as q, that is, $n_G \cdot G = G + G + \cdots + G$ (n_G times) $= \mathcal{O}$, the point at infinity or zero point. For all the points $A = (x_A, y_A)$, $B = (x_B, y_B) \in E_q(\kappa, \mu)$, the elliptic curve point addition and elliptic curve point (scalar) multiplication are defined as follows.

Elliptic Curve Addition: The point $C = A + B = (x_C, y_C)$ is determined by $x_C = (\lambda^2 - x_A - x_B) \pmod q$ and $y_C = (\lambda(x_A - x_C) - y_A) \pmod q$, where

$$\lambda = \begin{cases} \frac{y_B - y_A}{x_B - x_A} \pmod q, & \text{if } A \neq -B \\ \frac{3x_A^2 + \kappa}{2y_A} \pmod q, & \text{if } A = B \end{cases}$$

Elliptic Curve Multiplication: Scalar multiplication of an elliptic curve point $A = (x_A, y_A)$ with a scalar l denoted as $l.A$ is defined using repeated additions as $l.A = A + A + \cdots + A$ (l times). The number of additions to be performed is reduced by using point doubling operations.

Step S_2*:* The TRA picks a "collision-resistant one-way cryptographic hash function", say $H(\cdot)$ (for instance, "Secure Hash Standard (SHA-256) hash algorithm may be used).

Step S_3*:* The TRA chooses a private key pr_{TRA} in $Z_q^* = \{1, 2, \cdots, q-1\}$ for itself and computes the public key $Pub_{TRA} = pr_{TRA}.\ G$, and publishes Pub_{TRA} and domain parameters $\{E_q(\kappa, \mu),\ G,\ H(\cdot)\}$ as public.

C. Registration Phase

The TRA executes this phase to register each entity individually through a dedicated registration phase.

1) IoT Smart Device Registration Phase: This phase allows the TRA to register the IoT Smart Sensor (SN) using secure channel.

Step SDR_1*:* TRA picks ID_{SND}, $s \in Z_q^*$, RTS_S and computes the pseudo-identity as $RID_{SND} = H(ID_{SND} || s || RTS_S || pr_{TRA})$ and the temporary identity as $TID_{SND} = H(RID_{SND} || s || pr_{TRA} || RTS_S)$. TRA picks the private key as $pr_S \in Z_q^*$ and the corresponding public key as $Pub_S = pr_S \cdot G$.

Step SDR_2*:* TRA pre-loads SN with $\{(RID_{SND}, TID_{SND}), H(\cdot), E_q(\kappa, \mu), G, (pr_S, Pub_S)\}$.

2) Mobile Vehicle Registration Phase: This phase allows the TRA to register the Mobile Vehicle (MV) using a secure channel.

Step MVR_1*:* MV picks its private identity ID_M and forwards it to the TRA via a secure channel. TRA picks a private random secret $m \in Z_q^*$ and generates a timestamp for identity generation in registration RTS_m. TRA then computes the pseudo-identity $RID_M = H(ID_M || m || RTS_m || pr_{TRA})$ and the temporary identity $TID_M = H(RID_M || m || pr_{TRA} || RTS_m)$. TRA sends RID_M, TID_M back to the MV via the secure channel.

Step MVR_2*:* MV picks its private key as pr_M and the corresponding public key as $Pub_M = pr_M \cdot G$. The mobile vehicle then publishes Pub_M as its public key and sends Pub_M to TRA via the secure channel.

Step MVR_3*:* TRA generates a private session key parameter for MV as $K_M = H(RID_M || Pub_M || pr_{TRA} || ID_M || m)$. TRA then associates the MV with its FS by generating an association key K_{MV_i, FS_j} or retrieving the association key from the blockchain if it already exists. It also generates

a timestamp TS_{mc} and computes $K_M^* = H(K_M||TS_{mc}) \oplus H(K_{MV_i,FS_j}||\ TID_M||\ RID_M||TS_{mc})$ that hides the MV's contribution to the session key with FS.

Step MVR_4: TRA creates a transaction $Tx_i = \langle TID_M, K_M^*, E_{Pub_M}(ID_M, K_{MV_i,FS_j}, RID_M, TS_{mc})\rangle$, signs the transaction with $Sig_{Tx_i} = ECDSA.sig_{pr_{TRA}}(Tx_i)$ using the signature method $sig(\cdot)$ of the "elliptic curve digital signature algorithm (ECDSA)" with the private key pr_{TRA} and forwards $\langle Tx_i, Sig_{Tx_i}\rangle$ to the fog server FS via a secure channel. The leader fog server creates AuthCred block with its collected n_m transactions, executes the "Practical Byzantine Fault Tolerance (PBFT)" described in [44] (see Algorithm 1) within the same fog system, followed by mining and addition of the block into the blockchain. An MV is associated with a single FS only.

3) Fog Server Registration Phase: This phase allows the TRA to register a fog server (FS).

Step FSR_1: FS picks its private identity as ID_F and forwards it to the TRA via a secure channel. TRA picks its private random secret $f \in Z_q^*$ and generates a timestamp RTS_f. TRA then computes the pseudo-identity $RID_F = H(ID_F||\ f||\ RTS_f||\ pr_{TRA})$ and the temporary identity $TID_F = H(RID_F||\ f||\ pr_{TRA}||\ RTS_f)$. TRA sends RID_F, TID_F back to the mobile vehicle MV via the secure channel.

Step FSR_2: FS picks its private key pr_F and the corresponding public key as $Pub_F = pr_F \cdot G$. The FS then publishes Pub_F as its public key and sends Pub_F to TRA via a secure channel.

Step FSR_3: TRA generates a private session key parameter for FS as $K_F = H(RID_F||\ Pub_F||\ pr_{TRA}||\ ID_F||f)$. TRA then associates the FS with all its MV by generating an association key K_{MV_i,FS_j} or retrieving the association key from the blockchain if it already exists. It also generates a timestamp TS_{fc} and computes $K_F^* = H(K_F||TS_{fc}) \oplus H(K_{MV_i,FS_j}||\ TID_F||\ RID_F||TS_{fc})$ that hides the FS's contribution to the session key with MV.

Step FSR_4: TRA creates a transaction $Tx_j = \langle TID_F, K_F^*, E_{Pub_F}(ID_F, K_{MV_i,FS_j}, RID_F, TS_{fc})\rangle$, signs this transaction with its private key pr_{TRA} to obtain $Sig_{Tx_j} = ECDSA.sig_{pr_{TRA}}(Tx_j)$ and forwards $\langle Tx_j, Sig_{Tx_j}\rangle$ to a fog server in the blockchain for mining. The leader fog server creates an AuthCred block with its collected n_f transactions, executes the consensus algorithm (see Algorithm 1) within the same fog system, followed by mining and adds the block to the blockchain. An FS may be associated with multiple MVs, and the different association keys for a given FS are identified by the corresponding TID_M of the MV as stored in the transaction.

D. Authentication Phase

In this section, we consider two types of authentication mechanisms: 1) between an IoT smart device (SN) and a mobile vehicle (MV), called SNMV authentication and 2) between a mobile vehicle (MV) and a fog server (FS), called MVFS authentication.

1) Authentication Between IoT Smart Device and Mobile Vehicle: The following steps explain this phase (SNMV authentication phase):

Step $SNMV_1$: SN picks a private random secret $i_S \in Z_q^*$ along with a timestamp TS_S. It computes the corresponding public parameter as $I_S = H(i_S||\ TID_{SND}||\ RID_{SND}||\ pr_S||\ TS_S) \cdot G$ and a signature on the message as $Sig_S = H(i_S||\ TID_{SND}||\ RID_{SND}||\ pr_S||\ TS_S) + H(RID_{SND}||\ Pub_S||\ TID_{SND}||\ TS_S)\ *\ pr_S \pmod q$. The sensor SN sends the message Msg_{SM_1}: $\langle I_S, TID_{SND}, RID_{SND}, TS_S, Sig_S\rangle$ to the mobile vehicle MV.

Step $SNMV_2$: The mobile vehicle MV receives the message Msg_{SM_1} at the time TS_S^* and verifies the timestamp as $|TS_S^* - TS_S| \leq \Delta T$. If it is verified as true, the signature Sig_S is verified. The signature is verified as $Sig_S \cdot G \stackrel{?}{=} I_S + H(RID_{SND}||\ Pub_S||\ TID_{SND}||\ TS_S) \cdot Pub_S$.

Step $SNMV_3$: MV picks a private random secret $j_M \in Z_q^*$ and a timestamp TS_M to compute the public parameter $J_M = H(j_M||\ TID_M||\ RID_M||\ pr_M||\ TS_M) \cdot G$. The session key between the sensor and the mobile vehicle is computed as $SK_{MVS} = H(j_M||\ TID_M||\ RID_M||\ pr_M||\ TS_M) \cdot I_S$. A signature is generated over the private random j_M and the session key as $Sig_M = H(j_M||\ TID_M||\ RID_M||\ pr_M||\ TS_M) + H(J_M||\ SK_{MVS}||\ TID_{SND}||\ TS_M)\ *\ pr_M \pmod q$. MV picks a new temporary session identity for the smart device $TID_{SND}^{new} \in Z_q^*$ and hides it by computing $TID_{SND}^* = TID_{SND}^{new} \oplus H(TID_{SND}||\ SK_{MVS}||\ TS_M||\ Sig_M)$. MV sends Msg_{SM_2} : $\langle J_M, Sig_M, TID_M, RID_M, TID_{SND}^*, TS_M\rangle$ to SN.

Step $SNMV_4$: SN receives the message Msg_{SM_2} at the timestamp TS_M^* and verifies it as $|TS_M^* - TS_M| \leq \Delta T$. It then computes the session key as $SK_{SMV} = H(i_S||\ TID_{SND}||\ RID_{SND}||\ pr_S||\ TS_S) \cdot J_M$ and checks the signature Sig_M as $Sig_M \cdot G \stackrel{?}{=} J_M + H(J_M||\ SK_{SMV}||\ TID_{SND}||\ TS_M) \cdot Pub_M$. If the signature verification is successful, then it extracts $TID_{SND}^{new} = TID_{SND}^* \oplus H(TID_{SND}||\ SK_{SMV}||\ TS_M||\ Sig_M)$ and updates its current temporary identity TID_{SND} with new temporary identity TID_{SND}^{new}. It then generates a new temporary identity for MV as $TID_M^{new} \in Z_q^*$ and a timestamp TS_{SM}. It hides TID_M^{new} as $TID_M^* = TID_M^{new} \oplus H(TID_M||\ SK_{SMV}||\ TS_{SM})$ and computes the session key verifier as $SKV_{SMV} = H(SK_{SMV}||\ TS_{SM}||\ TID_M^{new})$ and sends the message Msg_{SM_3}: $\langle TID_M^*, SKV_{SMV}, TS_{SM}\rangle$ to the mobile vehicle.

Step $SNMV_5$: After the timestamp is verified correctly as $|TS_M^* - TS_M| \leq \Delta T$ using Msg_{SM_2} received at TS_M^*, MV extracts TID_M^{new} from $TID_M^{new} = TID_M^* \oplus H(TID_M||\ SK_{MVS}||\ TS_{SM})$ and computes the session key verifier as $SKV_{MVS} = H(SK_{MVS}||\ TS_{SM}||\ TID_M^{new})$. If the received SKV_{SMV} is equal to the computed SKV_{MVS}, it stores the session key SK_{MVS} and updates TID_M with TID_M^{new}. SN also stores the session key SK_{SMV} in its memory for the later communication with MV.

This phase is summarized in Fig. 3.

2) Authentication Between Mobile Vehicle and Fog Server: This MVFS authentication phase establishes a session key between a mobile vehicle (MV) and its associated fog server (FS) after mutual authentication upon retrieving the registered credentials from the blockchain during registration process.

Step $MVFS_1$: MV requests the fog server FS for the transaction containing TID_M from the blockchain.

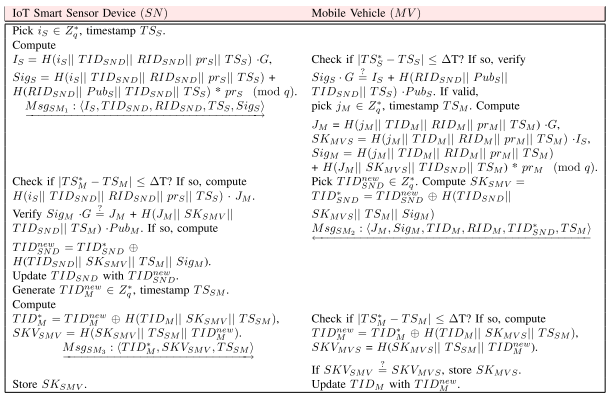


Fig. 3. Summary of authentication phase between SN and MV.

FS retrieves and sends $Tx_i = \langle TID_M, K_M^*, E_{Pub_M}(ID_M, K_{MV_i,FS_j}, RID_M, TS_{mc}) \rangle$ to the MV.

Step $MVFS_2$: MV decrypts TX_i using the private key pr_M to obtain $\langle ID_M, K_{MV_i,FS_j}, RID_M, TS_{mc} \rangle$. It then generates a private random secret $u_M \in Z_q^*$ and a timestamp TS_{mf1}, and hides RID_M as $RID_M^* = RID_M \oplus H(K_{MV_i,FS_j}||\ TS_{mf1})$ and computes $U_M = H(u_M||\ pr_M||\ TID_M||\ TS_{mf1}) \cdot G$. TRA's contribution to the session key for MV is extracted from K_M^* as $H(K_M||\ TS_{mc}) = K_M^* \oplus H(K_{MV_i,FS_j}||\ TID_M||\ RID_M||\ TS_{mc})$. It is hidden as $K_M^h = H(K_M||\ TS_{mc}) \oplus H(RID_M||\ TID_M||\ Pub_F||\ K_{MV_i,FS_j}||\ TS_{mf1})$. It computes a signature over u_M as $Sig_M = H(u_M||\ pr_M||\ TID_M||\ TS_{mf1}) + H(TID_M||\ RID_M||\ Pub_M||\ Pub_F||\ H(K_M||\ TS_{mc})||\ TS_{mf1}) * pr_M \pmod q$. MV sends the message $Msg_{MF_1} = \langle U_M, TID_M, RID_M^*, Sig_M, K_M^h, TS_{mf1} \rangle$ to FS.

Step $MVFS_3$: FS verifies the timestamp of the message received at TS_{mf1}^* as $|TS_{mf1}^* - TS_{mf1}| \leq \Delta$ T. It extracts RID_M from RID_M^* using association key K_{MV_i,FS_j} as $RID_M = RID_M^* \oplus H(K_{MV_i,FS_j}||\ TS_{mf1})$. FS extracts $H(K_M||\ TS_{mc}) = K_M^h \oplus H(RID_M||\ TID_M||\ Pub_F||\ K_{MV_i,FS_j}||\ TS_{mf1})$ and verifies the signature as $Sig_M \cdot G \stackrel{?}{=} U_M + H(TID_M||\ RID_M||\ Pub_M||\ Pub_F||\ H(K_M||\ TS_{mc})||\ TS_{mf1}) \cdot Pub_M$.

Step $MVFS_4$: FS generates a timestamp TS_{mf2} if Sig_M is verified to be correct. It retrieves the block with transaction containing TID_F as $Tx_j = \langle TID_F, K_F^*, E_{Pub_F}(ID_F, K_{MV_i,FS_j}, RID_F, TS_{fc}) \rangle$. It uses its own private key pr_F to decrypt and obtain $\langle ID_F, K_{MV_i,FS_j}, RID_F, TS_{fc} \rangle$. The extracted RID_F is hidden as $RID_F^* = RID_F \oplus H(K_{MV_i,FS_j}||\ TS_{mf2})$. $TRA's$ session key contribution for FS is extracted from K_F^* as $H(K_F||TS_{fc}) = K_F^* \oplus H(K_{MV_i,FS_j}||\ TID_F||\ RID_F||\ TS_{fc})$ and hidden as $K_F^h = H(K_F||TS_{fc}) \oplus H(RID_F||\ TID_F||\ Pub_M||\ K_{MV_i,FS_j}||\ TS_{mf2})$.

Step $MVFS_5$: FS generates $v_F \in Z_q^*$ and computes $V_F = H(v_F||\ pr_F||\ TID_F||\ TS_{mf2}) \cdot G$, the Diffie-Hellman type key as $DK_{FM} = H(v_F||\ pr_F||\ TID_F||\ TS_{mf2}) \cdot U_M$ and finally the session key with MV as $SK_{FM} = H(H(K_M||\ TS_{mc})||\ U_M||\ V_F||\ H(K_F||TS_{fc})||\ DK_{FM})$. A signature is generated over v_F as $Sig_F = H(v_F||\ pr_F||\ TID_F||\ TS_{mf2}) + H(SK_{FM}||\ DK_{FM}||\ RID_M||\ RID_F||\ H(K_F||TS_{fc})\ ||\ TS_{mf2}) * pr_F \pmod q$. MV generates a new temporary identity as $TID_M^{new} \in Z_q^*$ and hides it as TID_M^*. The message $Msg_{MF_2} : \langle V_F, Sig_F, TID_M^*, TID_F, RID_F^*, K_F^h, TS_{mf2} \rangle$ is then sent to the MV.

Step $MVFS_6$: MV receives Msg_{MF_2} at TS_{mf2}^* and verifies the timestamp as $|TS_{mf2}^* - TS_{mf2}| \leq \Delta$T. It extracts $RID_F = RID_F^* \oplus H(K_{MV_i,FS_j}||\ TS_{mf2})$ and. TRA's contribution for FS is extracted as $H(K_F||TS_{fc}) = K_F^h \oplus H(RID_F||\ TID_F||\ Pub_M||\ K_{MV_i,FS_j}||\ TS_{mf2})$. It then computes the Diffie-Hellman parameter $DK_{MF} = H(u_M||\ pr_M||\ TID_M||\ TS_{mf1}) \cdot V_F$ and the session key as $SK_{MF} = H(H(K_M||\ TS_{mc})||\ U_M||\ V_F||\ H(K_F||TS_{fc})||\ DK_{MF})$. It then verifies FS's signature on Msg_{MF_2} as $Sig_F \cdot G \stackrel{?}{=} V_F + H(SK_{FM}||\ DK_{MF}||\ RID_M||\ RID_F||\ H(K_F||TS_{fc})||\ TS_{mf2}) \cdot Pub_F$. If so, it extracts TID_M^{new} from TID_M^* and updates TID_M with TID_M^{new} in its database.

Step $MVFS_7$: MV creates a new temporary session identity for FS as $TID_F^{new} \in Z_q^*$ and hides it as TID_F^*. It then generates a new timestamp TS_{mf3} and computes a session key verifier $SKV_{mf} = H(SK_{mf}||\ TS_{mf}||\ TID_F^{new})$. The final message for this phase is sent from MV to FS as $Msg_{MF_3} : \langle SKV_{mf}, TS_{mf3}, TID_F^* \rangle$.

Step $MVFS_8$: FS receives Msg_{MF_3} at time TS_{mf3}^*, verifies the timestamp by the condition: $|TS_{mf3}^* - TS_{mf3}| \leq \Delta$T and extracts TID_F^{new} from TID_F^*. It then computes the session key verifier as $SKV_{fm} = H(SK_{fm}||\ TS_{mf3}||\ TID_F^{new})$. If the computed SKV_{fm} and received SKV_{mf} match, then it stores SK_{fm} in memory and updates TID_F with TID_F^{new} in its database. MV then stores SK_{mf} in memory as the session key for further communication exchange between MV and FS.

The summary of the MVFS authentication phase is provided in Fig. 4.

E. Secure Data Aggregation With Block Creation, Verification and Addition in BC

This section provides a compendious presentation of the creation of transactions by the TRA and the creation of blocks by the fog server FS with the following steps.

Step $BCFS_1$: The TRA creates transactions $Tx_i = \langle TID_M, K_M^*, E_{Pub_M}(ID_M, K_{MV_i,FS_j}, RID_M, TS_{mc}) \rangle$ and $Tx_j = \langle TID_F, K_F^*, E_{Pub_F}(ID_F, K_{MV_i,FS_j}, RID_F, TS_{fc}) \rangle$ for the mobile vehicle MV and the fog server FS, respectively, during registration as shown in Section IV-C.2 and IV-C.3, respectively. TRA signs the transactions with the ECDSA signature generation algorithm $sig(\cdot)$ using pr_{TRA} as $Sig_{Tx_i} = ECDSA.sig_{pr_{TRA}}(Tx_i)$ and $Sig_{Tx_j} = ECDSA.sig_{pr_{TRA}}(Tx_j)$. TRA sends $\langle Tx_i, Sig_{Tx_i} \rangle$ to the FS associated with the respective MV and also sends $\langle Tx_j, Sig_{Tx_j} \rangle$ to the registering fog server FS. It is to be noted that only part of each transaction is encrypted.

Step $BCFS_2$: The fog server FS verifies the signatures Sig_{Tx_i} and Sig_{Tx_j} for Tx_i and Tx_j, respectively. If the signatures are valid, the FS marks them as valid. After the FS receives n_m transactions from the TRA, it creates a block $Block_m$ with a message $BlockMsg_m = (Tx_{i1}\ ||Tx_{i2}\ ||\cdots\ ||Tx_{in_m})$ and a signature on the block as $Sig_{pr_F}(BlockMsg_m)$. Similarly, after the FS receives n_f transactions from the TRA, it creates a block $Block_f$ with the message as $BlockMsg_f = (Tx_{j1}\ ||Tx_{j2}\ ||\cdots||Tx_{jn_f})$

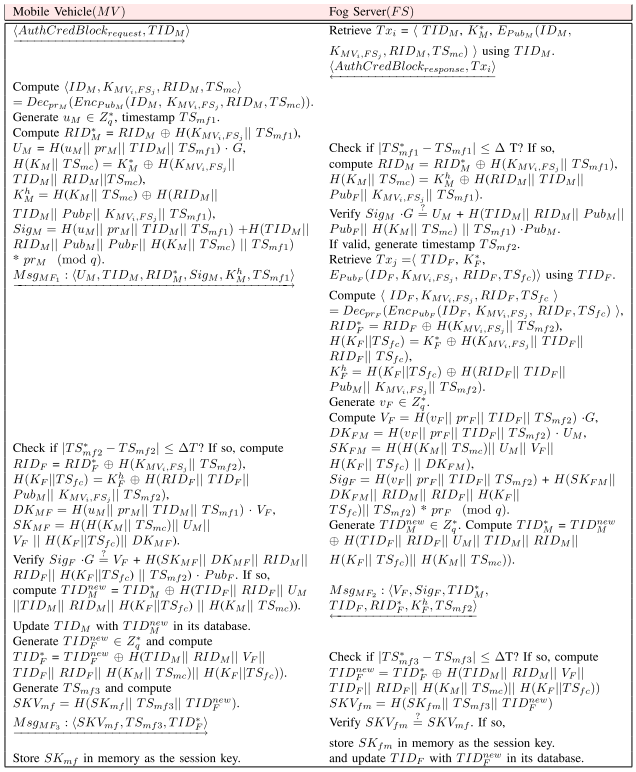


Fig. 4. Summary of authentication phase between MV and FS.

and a signature on the block as $Sig_{pr_F}(BlockMsg_f)$. Such created blocks have the formats as shown in Fig. 5 with the BlockType designated as "$AuthCred_m$" and "$AuthCred_f$" for mobile vehicles and fog servers, respectively. Once a block is created by the fog server, the following tasks are performed: a) the fog system executes a leader selection algorithm to select a fog server as the leader using [45], and b) a consensus algorithm is executed to validate the block and add the block to the blockchain using Algorithm 1.

Step $BCFS_3$: During the $MVFS$ phase of the authentication scheme as shown in Section IV-D.2, the blockchain is searched for the block containing the transaction with the required TID_M or TID_F, and the AuthCred block is retrieved from the chain and sent to the requesting entity. Based on this retrieved block from the chain, the rest of the steps in the $MVFS$ phase proceeds. If the scheme succeeds in executing all its steps, a session key is established between MV and FS.

Step $BCFS_4$: The mobile vehicle MV collects sensitive sensor data $SNDATA$ from the IoT smart sensor devices SN corresponding to its associated zone in the farm field. This data is encrypted by SN using the session key SK_{SMV} established in $SNMV$ phase as shown in Section IV-D.1. The encrypted data is sent to the mobile vehicle MV_i. The mobile vehicle MV_i decrypts the data with SK_{MVS}. MV_i then encrypts $SNDATA$ with SK_{MF} and sends it to the fog server FS_j, which decrypts using SK_{FM}.

Step $BCFS_5$: The fog server FS_j creates a transaction $Tx_d = \langle ID_F, TS_F, Z_i, SNDATA \rangle$ for each received sensor data $SNDATA$ and creates a "SensorData" block $Block_d$ as shown in Fig. 6 with the n_d collected transactions from $BlockMsg_d = (Enc_{Pub_F}(Tx_{d1}) || Enc_{Pub_F}(Tx_{d2}) || \cdots || Enc_{Pub_F}(Tx_{dn_d}))$ and a signature on the block as $Sig_{pr_F}(BlockMsg_d)$. The BlockType is designated as "SensorData". After the creation of the block, the fog servers in the fog system select a leader and execute the consensus algorithm provided in Algorithm 1 to validate the block and add it into the blockchain.

Algorithm 1 Achievement of Consensus of the Blockchain

Input: $Block_k$: A full block with the structure as given in Fig. 5 or Fig. 6 that is to be added to the blockchain, N_f: Total number of P2P nodes (fog servers) in the blockchain network where $k = m$ for $Block_m$ or $k = f$ for $Block_f$ or $k = d$ for $Block_d$, $Block_k$ has n_m or n_f or n_d transactions according to the block type being added
Output: Status of block commit operation (YES/NO)

1: Select a leader FS_L among the fog nodes
2: Set $Limit_{Votes} = 2 * N_{faulty} + 1$
3: $AllFogVotes \leftarrow$ NIL
4: FS_L broadcasts $Block_k$ to all fog nodes in a Peer-to-Peer (P2P) network
5: **for** each fog server node FS_j in the P2P network **do**
6: Set $Fog\_Vote_j = NO$
7: Compute $Block\_Hash = H(Block_k)$
8: **if** ($Block\_Hash = CBHash$) **then**
9: **if** (block signature is valid) **then**
10: Create Merkle tree root (MTR_{Block}) with n_k transactions from block payload
11: **if** (MTR_{Block} matches with the MTR in the block) **then**
12: Set $Fog\_Vote_j = YES$
13: **end if**
14: **end if**
15: **end if**
16: FS_j encrypts Fog_Vote_j using the public key Pub_{F_L} of FS_L as $ECCEnc_{Pub_{F_L}}(Fog\_Vote_j)$
17: Add $ECCEnc_{Pub_{F_L}}(Fog\_Vote_j)$ to $AllFogVotes$ and send it to FS_L
18: **end for**
19: Set $Vote_{Count} \leftarrow 0$
20: **for** each encrypted vote V_t reply in $AllFogVotes$ **do**
21: FS_l computes $V_t = ECCDec_{pr_{F_L}}(ECCEnc_{Pub_{F_L}}(V_t))$, where $V_t = Fog\_Vote_j$
22: **if** (V_t is valid) **then**
23: Set $Vote_{Count} = Vote_{Count} + 1$
24: **end if**
25: **end for**
26: **if** ($Vote_{Count} \geq Limit_{Votes}$) **then**
27: FS_l adds block $Block_k$ into its blockchain
28: FS_L broadcasts block commit status as YES to the blockchain network
29: Other peer fog nodes add the block into their blockchains
30: **end if**

To analyse the storage space and communication cost, the following notations are used. The identities and random secrets are considered to be of 160 bits each. The length of output of hash function, cipihertext block of "symmetric key encryption/decryption using the Advanced Encryption Standard (AES-128) encryption algorithm" are taken as 256 bits and 128 bits, respectively. For public key cryptographic operations, the "Elliptic Curve Cryptosystem (ECC)" is chosen such that 160-bit ECC provides the same security level as that for 1024-bit RSA cryptosystem [46]. An elliptic curve point $A = (x_A, y_A)$ needs (160+160) = 320 bits, and the timestamp requires 32 bits. To analyse the computational operations, the following notations are also used. T_h stands for the hash operation performed using SHA-256 hash algorithm, T_{ecm}

Block Header	
Block Version ($BVer_{ac}$)	Unique serial number
Previous Block Hash (PBH_{ac})	Hash value of previous block
BlockType	Authentication Credentials ($AuthCred_m$ or $AuthCred_f$)
Merkle Tree Root (MTR_{ac})	Merkle tree root on transactions
Timestamp (TS_{ac})	Block creation time
Owner of Block OWN_{ac}	Fog server (FS_j)
Public key of transactions verification	Pub_{CRA}
Public key of block signer	Pub_F
Block Payload (Transactions)	
MV Transactions Tx_i	$\{(Tx_{il}, ECDSA.sig_{Tx_{il}}) \mid l = 1, 2, \cdots, n_m\}$
OR	$\vdots$
FS Transactions Tx_j	$\{(Tx_{jl}, ECDSA.sig_{Tx_{jl}}) \mid l = 1, 2, \cdots, n_f\}$
ECDSA signature on Block	$Sig_{pr_F}(BlockMsg_m)$ or $Sig_{pr_F}(BlockMsg_f)$
Current Block Hash ($CBHash_{ac}$)	Hash value of present block

Fig. 5. Structure of an AuthCred block $Block_m$ or $Block_f$ in the blockchain.

and T_{eca} represent the elliptic curve multiplication and addition, respectively. $T_{SymmEnc}$ and $T_{SymmDec}$ denote the AES-128 symmetric encryption/decryption operations, respectively. T_{CKDf} represents the cryptographic key derivation function, which is considered as T_{ecm}. The time for signature generation using $ECDSA$ is $T_h + T_{ecm}$ while the verification time using $ECDSA$ takes $2T_{ecm} + T_{eca} + T_h$. The time T_{ECCEnc} for ECC encryption takes $2T_{ecm} + T_{eca}$, while the time T_{ECCDec} for ECC decryption needs $T_{ecm} + T_{eca}$ operations. Additionally, T_{mul} and T_{add} represent "modular multiplication" and "modular addition" over the finite field $GF(q)$, respectively.

The size of AuthCred block as shown in Fig. 5 is computed with the sizes of the components $\{BVer_{ac}, PBH_{ac}, BlockType, MTR_{ac}, TS_{ac}, OWN_{ac}, Pub_{TRA}, Pub_F, \{(Tx_i, ECDSA.sig_{Tx_i})\}_{i=1,2,\cdots,n_m}, Sig_{pr_F}(BlockMsg_m), CBHash\}$ as 32, 256, 32, 256, 32, 160, 320, 320, $1472 * n_m$, 320, and 256 bits, respectively. Therefore, the total size of AuthCred block becomes $1984 + 1472n_m$ bits. Similarly, the size of AuthCred block for the FS point of view shown in Fig. 5 is computed with the sizes of $\{BVer_{ac}, PBH_{ac}, BlockType, MTR_{ac}, TS_{ac}, OWN_{ac}, Pub_{TRA}, Pub_F, \{(Tx_j, ECDSA.sig_{Tx_j})\}_{j=1,2,\cdots,n_f}, Sig_{pr_F}(BlockMsg_f), CBHash\}$ as 32, 256, 32, 256, 32, 160, 320, 320, $1472 * n_f$, 320, and 256 bits, respectively. Therefore, the total size of the AuthCred block becomes $1984 + 1472n_f$ bits for blocks with transactions from FS. The size of SensorData block in Fig. 6 is computed with the sizes of $\{BVer_d, PBH_d, BlockType, MTR_d, TS_d, OWN_d, Pub_F, \{(E_{Pub_F}(Tx_d), ECDSA.sig_{Tx_d})\}_{d=1,2,\cdots,n_d}, Sig_{pr_F}(BlockMsg_d), CBHash\}$ as 32, 256, 32, 256, 32, 160, 320, $640 * n_d$, 320, and 256 bits, respectively, and the total size becomes $1664 + 640n_d$ bits.

The proposed consensus Algorithm 1 is vote-based with threshold in "Practical Byzantine Fault Tolerance (PBFT)" as described in [44]. A leader fog node broadcasts a block $Block_k$ to the follower fog nodes and collects their vote on its validity for addition to the blockchain. Every fog server validates the block hash, block signature, and Merkle tree root and prepares its vote. The fog server encrypts its vote with the leader's public key using ECC encryption before sending the response in the public channel. The leader decrypts each vote using ECC decryption algorithm. If the vote favors block addition, the leader increments the favorable vote count. The validated

Block Header	
Block Version	$BVer_d$
Previous Block Hash	PBH_d
BlockType	Sensing Data ($SensorData$)
Merkle Tree Root	MTR_d
Timestamp	TS_d
Owner of Block OWN_d	FS_j
Public Key of Signer FS	Pub_F
Block Payload (Encrypted Transactions)	
List of n_d Encrypted Transactions $\#l$ (Tx_{dl})	$\{(E_{Pub_F}(Tx_{dl}), ECDSA.sig_{Tx_{dl}}) \mid l = 1, 2, \cdots, n_d\}$
ECDSA Signature on Block	$Sig_{pr_F}(BlockMsg_d)$
Current Block Hash	$CBHash_d$

Fig. 6. Structure of a SensorData block $Block_d$.

block is added to the chain when the total positive votes are greater than the fault limit.

1) Computational Complexity: Any step in Algorithm 1 that is directly linked to a change in the blockchain is considered on-chain. If a block fails verification or does not receive the required votes, it will not be added to the chain and thus, it cannot affect the blockchain state.

- **Off-chain**: Steps 1 to 26 in the Algorithm 1 are executed off-chain. The verification of the block hash requires T_h operations. The Merkle tree root contains N_k nodes and takes $O(\log_2 N_k)$ operations. Thus, the total computational complexity for N_f fog nodes is at least $N_f * (2T_h + 5T_{ecm} + 3T_{eca} + O(\log_2 N_k))$.
- **On-chain**: Steps 27 and 28 in the Algorithm 1 are executed on-chain. Adding a block to the blockchain consists of finding the most recent block and linking the new block to the chain. As these steps take constant time, the on-chain computational complexity is constant.

The total computational complexity is denoted and estimated by $C_{comp} = N_f * (2T_h + 5T_{ecm} + 3T_{eca} + O(\log_2 N_k)) + O(1)$.

2) Communication Complexity: The size of AuthCred block is $(1984 + 1472n_m)$ and SensorData block is $(1664 + 640n_d)$ bits. Note that $160 * N_f$ bits are communicated for the encrypted votes from N_f fog servers. Thus, The algorithm uses $N_f * (1984 + 1472n_m) + N_f$ (160) bits for communicating the AuthCred block and $N_f * (1664 + 640n_d) + N_f$ (160) bits to communicate SensorData block.

3) Fault Tolerance: Let N_{fault} be the number of faulty fog servers in a fog system with N_f servers. A consensus is reached even if only N_f - N_{fault} servers communicate among themselves. However, the unresponsive N_{fault} fog servers might not be faulty, and the responsive N_{fault} fog servers might be faulty. Even with these cases a consensus is attainable if responses from the non-faulty fog servers outnumber the responses from the faulty ones under the condition N_f - $2N_{fault} > N_{fault}$ i.e $N_f > 3N_{fault}$. The fog system achieves a consensus if the system has at least $N_f = 3N_{fault} + 1$ fog servers. Out of these, when $2N_{fault} + 1$ servers reach a consensus, the fog system reaches a consensus too. In other words, out of any number of N_f servers, if $1/3^{rd}$ nodes are faulty, the consensus is declared for positive responses from $2/3^{rd}$ servers, that is, when positive votes are $2(N_f - 1)/3 + 1$.

4) Blockchain Retrieval Time: Let len_{BC} be the length of the blockchain at a time when the Big Data Centre requests data from the blockchain. The blocks are traversed using the previous block hash, which requires T_h time to compute the

block hash. Thus, the time to access the blockchain is given by $T_h * len_{BC}$.

5) Latency and Energy Consumption: The proposed consensus algorithm does not solve any hashing puzzles. Instead, similar to PBFT, it uses regular communication to reach a consensus. It is highly efficient for small-scale networks. Let L_{block} be the transmission latency of broadcasting a new block to all nodes, ET be the effective throughput, L_{Vote} be the transmission latency of vote responses, L_V be the latency of block verification, $size_{block}$ is the block size, $size_{Vote}$ be the size of a vote response message. Then the transmission latency of block broadcast and vote responses are given as $L_{block} = size_{block}/ET * N_f^2$ and $L_{Vote} = size_{Vote}/ET * N_f^2$, respectively. If P_{Tr} is the transmission power, then the energy consumption for transmission is given as $P_{Tr} * (L_{block} + L_{Vote}) * N_f^2 = P_{Tr} * (size_{block}/ET + size_{Vote}/ET) * N_f^2$. If P_{Comp} is the computational power, then the energy consumption for computation is given as $P_{Comp} * L_V$, where L_V is given as $size_{block}/Cap_{CPU}$, with C_{CPU} being the CPU capacity of the fog nodes. Thus, the total energy consumption of one new block with the proposed consensus Algorithm 1 is given as $P_{Tr} * N_f^2 * (size_{block}/ET + size_{Vote}/ET) + P_{Comp} * N_f * (size_{block}/Cap_{CPU})$ [47].

Remark 1: The issue of storing a huge amount of data on the blockchain and its effect on retrieval time is related to the problem of implementation of the hybrid blockchain in the proposed scheme. Applying efficient data structures to store the blockchain can lead to efficient retrievals. Merkle tree can be implemented as B-Merkle tree using modified polynomial commitment scheme with proofs based on element ordering giving small proof sizes and low tree heights [48], [49]. Tu et al. [50] suggests B+ tree to store the blockchain keeping the transactions on leaf nodes and the indexes on inner nodes. Redis Cache technology is used for fast indexing of block files. Their proposed retrieval algorithm has better retrieval efficiency. Feng et al. [51] proposed the more efficient BB+ tree with Bloom filter for large blocks which improves the retrieval time by roughly 40% and 43% using single and multiple features, respectively. Thus, usage of such efficient methods to store and retrieve from the blockchain will ensure that the access of the hybrid blockchain during authentication does not slow down the overall performance of the proposed scheme.

F. Dynamic Nodes Addition Phase

This phase is used whenever an IoT smart device is corrupted or damaged and needs to be replaced with a new node. The TRA registers the newly placed node, and assigns the appropriate credentials consisting of identities, and public-private key pairs required for authentication.

Step DNA_1: TRA picks ID_{SND}^{new}, $s^{new} \in Z_q^*$, a registration timestamp RTS_S^{new} and computes the pseudo-identity as $RID_{SND}^{new} = H(ID_{SND}^{new} || s^{new} || RTS_S^{new} || pr_{TRA})$ and the temporary identity as $TID_{SND}^{new} = H(RID_{SND}^{new} || s^{new} || pr_{TRA} || RTS_S^{new})$. TRA picks the private key as $pr_S^{new} \in Z_q^*$ and the corresponding public key as $Pub_S^{new} = pr_S^{new} \cdot G$.

Step DNA_2: TRA pre-loads SN with $\{(RID_{SND}^{new}, TID_{SND}^{new}), H(\cdot), E_q(\kappa, \mu), G, (pr_S^{new}, Pub_S^{new})\}$.

V. Security Analysis

In this section, the security strength of the proposed scheme is analyzed using the "Real-or-Random (ROR) model" [52] and "Random Oracle Model (ROM)" [53] in formal security analysis. In addition, the proposed scheme is proved to be insusceptible to many widely known attacks with a non-mathematical (informal) security analysis. The scheme is then simulated for formal security verification under the widely recognized "Automated Validation of Internet Security Protocols and Applications (AVISPA)" software validation tool [54] to verify its safety.

A. Formal Security Analysis

In the proposed scheme, the IoT smart device SN and the mobile vehicle MV that collects data from SN establish a session key $SK_{SMV} = (SK_{MVS})$ during SNMV phase (see Section IV-D.1). The mobile vehicle MV and the fog server FS to which the data is deposited establish another session key $SK_{MF} = (SK_{FM})$ during the $MVFS$ phase (see Section IV-D.2).

1) Random Oracle Model: We define the respective security model based on the works by Bellare et al. [55] and Wu et al. [56], for the proposed scheme through a sequence of the interactive games between a challenger and an adversary. We prove that the proposed scheme provides the session key security against the adversary. For this purpose, the security model to analyze $AgroMobiBlock$ is defined in the supplementary material.

2) Provable Security: In this section, we now apply the random oracle model discussed in the supplementary material to prove that the scheme provides the session key security.

Theorem 1: Assume that a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ taking t_p execution time attempts to obtain the session key SK_{SMV} $(= SK_{MVS})$ established between smart device SN and mobile vehicle MV during the SNMV authentication phase, and the session key SK_{MF} $(= SK_{FM})$ established amid mobile vehicle MV and fog server FS for a given session during the MVFS authentication phase of the proposed $AgroMobiBlock$. If q_h, $|Hash|$ and $Adv_{\mathcal{A}}^{ECDDHP}(t_p)$ represent the "number of Hash queries", the "range space of a one-way collision-resistant hash function $H(\cdot)$" and the "advantage in breaking the Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP)", respectively, then $Adv_{\mathcal{A}}^{AgroMobiBlock}(t_p) \leq \frac{q_h^2}{|Hash|} + 2\ Adv_{\mathcal{A}}^{ECDDHP}(t_p)$.

Proof: The proof of this theorem is provided in the supplementary material. □

B. Informal Security Analysis

1) Replay Attack: The messages Msg_{SM_1}, Msg_{SM_2} and Msg_{SM_3} in SNMV phase and Msg_{MF_1}, Msg_{MF_2} and Msg_{MF_3} in MVFS phase are transmitted through open channels, whereas the timestamps TS_S, TS_M, TS_{SM} are used in SNMV phase and the timestamps TS_{mf1}, TS_{mf2}, TS_{mf3} are used in MVFS phase. In addition, the random secrets i_S, j_M, u_M and v_F, respectively, are used in the parameters I_S, J_M, U_M and V_F sent through the above messages. The receiver verifies each timestamp before processing the message contents. During

signature verification, the public parameters consisting of the random secrets are also verified. Due to this, the receiver can immediately identify if an adversary $\mathcal{A}$ captures a message and replays it. Thus, the proposed $AgroMobiBlock$ is resilient against replay attack.

2) Man-in-the-Middle (MiTM) Attacks: In the SNMV phase, any changes to the parameters I_S, TID_{SND}, and RID_{SND} in Msg_{SM_1}, J_M, TID_M, RID_M, and TID^*_{SND} in Msg_{SM_2}, and TID^*_M and SKV_{SMV} in Msg_{SM_3} are easily identified in the signature verification of Sig_S and Sig_M or session key verification of SKV_{SMV} as SKV_{MVS} at the receiver. Similarly, in the MVFS phase, any changes to the parameters U_M, TID_M, RID_M, and K^h_M in Msg_{MF_1}, V_F, TID_F, RID_F, and K^h_F in Msg_{MF_1}, and SKV_{mf} in Msg_{MF_1} can be easily identified in the signature verification of Sig_M and Sig_F or session key verification of SKV_{mf} as SKV_{fm} at the receiver. Thus, MiTM attack is resilient in the proposed $AgroMobiBlock$.

3) Impersonation Attacks: Consider the assumption that an adversary has intercepted the messages Msg_{SM_1}, Msg_{SM_2} and Msg_{SM_3} in the SNMV phase and Msg_{MF_1}, Msg_{MF_2} and Msg_{MF_3} in the MVFS phase. The following are the possible impersonation attacks:

• *IoT smart device impersonation attack*: Consider that the adversary $\mathcal{A}$ tries to impersonate the smart device SN in the SNMV phase. It has to fabricate a fake message $Msg^{adv}_{SM_1}$: $\langle I^{adv}_S, TID^{adv}_{SND}, RID^{adv}_{SND}, TS^{adv}_S, Sig^{adv}_S\rangle$ for which it has to generate I^{adv}_S and the timestamp TS^{adv}_S. However, this requires $\mathcal{A}$ to know the secrets i_S, ID_{SND}, and s. Since these secrets are never shared in any message, the IoT smart device impersonation attack is not possible in the proposed scheme.

• *Mobile vehicle impersonation attack*: Consider that $\mathcal{A}$ tries to impersonate the mobile vehicle MV in the SNMV and MVFS phases. The adversary $\mathcal{A}$ requires the knowledge of j_M, ID_M, and m to fabricate the message $Msg^{adv}_{SM_2}$: $\langle J^{adv}_M, Sig^{adv}_M, TID^{adv}_M, RID^{adv}_M, TID^{*adv}_{SND}, TS^{adv}_M\rangle$ in SNMV phase, and m, RID_M from TRA, u_M, ID_M from MV, along with K^*_M stored on the blockchain in order to fabricate the message $Msg^{adv}_{MF_1}$: $\langle U^{adv}_M, TID^{adv}_M, RID^{*adv}_M, Sig^{adv}_M, K^{h-adv}_M, TS^{adv}_{mf1}\rangle$ leading to conclude that the proposed scheme is resilient to mobile vehicle impersonation attack.

• *Fog server impersonation attack*: Consider that $\mathcal{A}$ tries to impersonate the fog server FS in the MVFS phase. It has to fabricate a fake message $Msg^{adv}_{MF_2}$: $\langle V^{adv}_F, Sig^{adv}_F, TID^{*adv}_M, TID^{adv}_F, RID^{*adv}_F, K^{h-adv}_F, TS^{adv}_{mf2}\rangle$ and requires the knowledge of the private secrets f, RID_F from TRA, ID_F, v_F from FS, along with K^*_F stored on the blockchain. Hence, $AgroMobiBlock$ is resilient against fog server impersonation attacks.

4) Privileged-Insider Attack: All the required secret credentials during the registration of the IoT smart device's ID_S, s, the mobile vehicle's ID_M, m, K_M and the fog server ID_F, f, K_F are either pre-loaded into the memory of the corresponding entity by the TRA or passed through a secure channel or not used directly. Thus, $AgroMobiBlock$ is strongly resilient against privileged insider attack.

5) Physical IoT Smart Device and Mobile Vehicle Capture Attacks: An adversary $\mathcal{A}$ who captures an IoT smart device SN can extract the information $\{(RID_{SND}, TID_{SND}), H(\cdot), E_q(\kappa, \mu), G, (pr_S, Pub_S)\}$ using power analysis attacks [57] and timing attacks [40]. A compromised smart device cannot affect the communication among the non-compromised smart nodes as none of the devices share any secret credentials or established session keys. Such a compromised node is replaced with a new node using the dynamic node addition phase as described in Section IV-F. Thus, $AgroMobiBlock$ is resilient against the physical IoT smart device capture attack. The adversary $\mathcal{A}$ cannot obtain any information from the captured mobile vehicle's memory as the blockchain stores its secret credentials securely. Hence, a physical capture attack on a mobile vehicle fails.

6) Ephemeral Secret Leakage (ESL) Attack: The session key in the SNMV phase, $SK_{SMV} = SK_{MVS}$, depends both on the temporal secrets i_S and j_M, and long term secrets TID_{SND}, RID_{SND}, TID_M, RID_M, pr_S and pr_M. Similarly, the session key in the MVFS phase, $SK_{FM} = SK_{MF}$, requires the dependency on the short-term secrets u_M and v_F, and the long term secrets K_M, K_F, TID_M, RID_M, TID_F, RID_F, pr_M and pr_F. We now consider the following two scenarios:

• If only the short-term secrets (i_S, j_M) in the SNMV phase and (u_M, v_F) in the MVFS phase are compromised, the session key $SK_{SMV} = SK_{MVS}$ is still uncompromised as $\mathcal{A}$ does not have access to the long term secrets (TID_{SND}, RID_{SND}, TID_M, RID_M, pr_S and pr_M) in SNMV phase and (K_M, K_F, TID_M, RID_M, TID_F, RID_F, pr_M and pr_F) in MVFS phase. Moreover, the secrets K_M, and K_F are only used in the hash computation and are never shared anywhere during the registration or authentication phase.

• If only the long term secrets (TID_{SND}, RID_{SND}, TID_M, RID_M, pr_S and pr_M) in SNMV phase and (K_M, K_F, TID_M, RID_M, TID_F, RID_F, pr_M and pr_F) in MVFS phase are compromised, the session key $SK_{SMV} = SK_{MVS}$ is still non-compromised as $\mathcal{A}$ does not have access to the short-term secrets are (i_S, j_M) in SNMV phase and (u_M, v_F) in MVFS phase.

For the adversary $\mathcal{A}$ to succeed in the ESL attack, the short-term and long-term secrets need to be compromised together. The proposed $AgroMobiBlock$ is then strongly resistant to ESL attacks under the CK adversary threat model.

7) Denial of Service (DoS) Attack: An attacker can launch a DoS attack to impede access to a server. To launch a DoS attack, the adversary accesses the server and consumes its resources to render them unusable for any other purpose. DoS attacks can be averted by preventing the adversary from gaining access to the server. After receiving Msg_{MF_1}, if verification of Sig_M fails at FS, the entity MV attempting to access FS will be deemed as not authentic and access to the fog server will be thwarted. Thus, $AgroMobiBlock$ is resistant to DoS attack.

8) Advanced Persistent Threat: An adversary slowly intrudes into the network to ex-filtrate specific targeted sensitive data using techniques, such as malware, social engineering, and zero-day vulnerabilities [58], [59], [60]. The proposed system uses the blockchain to actively store the authentication credentials and sensor data in separate blocks. The inherent capabilities of a blockchain including tamper-proof storage, secure the stored data. In addition, the

SensorData block stores the actual sensor data in an encrypted format. With no access to the private key of the fog server, the intruding group cannot access the sensor data as well.

9) Anonymity and Untraceability: The blockchain inherently possesses the characteristics of anonymity and untraceability on the data stored in the block. However, the authentication process must also satisfy the anonymity and untraceability properties separately. In the $SNMV$ phase, the messages Msg_{SM_1}, Msg_{SM_2}, and Msg_{SM_3} use only the temporary identities TID_{SND}, TID_M and hidden TID_M^*, and the pseudo-identities RID_{SND}, and RID_M instead of the original identities ID_{SND} and ID_M. Similarly, the $MVFS$ phase only uses the temporary identities TID_M and TID_F with the hidden pseudo-identities RID_M^* and RID_F^* instead of the original identities ID_M and ID_F. Thus, none of the messages can be traced back to the original identities of the sender. This preserves the anonymity of all the entities. The untraceability feature makes the public details in the messages unlinkable. Due to this, an attacker cannot obtain any information from the traffic in the public channel. Since none of the messages have any common contents, any parameters of a message cannot be inferred from other messages.

VI. Formal Security Verification Using AVISPA: Simulation Study

This section focuses on the simulation of the proposed scheme ($AgroMobiBlock$) using the widely regarded AVISPA tool [54] that validates the protocol is safe or unsafe by considering two cases: Case 1– authentication between SN and MV and Case 2– authentication between MV and FS. The simulation results presented in the supplementary material illustrate that the proposed scheme is safe against passive/active attacks, like replay and man-in-the-middle attacks.

VII. Comparative Study

This section compares the performance of the proposed $AgroMobiBlock$ on the costs of communication and computation for the SNMV authentication and key agreement phase (Phase 1) and the MVFS authentication and key agreement phase (Phase 2). These costs are studied against the schemes developed by Ali et al. [9], Chen et al. [10], Chae and Cho [11], Rangwani et al. [15], Wu and Tsai [16], Vangala et al. [17], Bera et al. [18], Tian et al. [23], Shuai et al. [24], Panda et al. [25], Eddine et al. [26], Fan et al. [27], Tomar and Tripathi [28], Itoo et al. [29] and Jia et al. [30] along with a thorough analysis of their security and functionality features.

A. Communication Costs Comparison

This section focuses on comparing the proposed scheme with the existing schemes in terms of the amount of cost expended in the communication of messages. The proposed $AgroMobiBlock$ scheme takes 2848 bits for communication in the SNMV phase and 4608 bits in the MVFS phase. Even with the exchanged transaction data in the first message of MVFS phase, the total communication cost only accounts for 4608 bits, which is reasonably comparable with respect to the other schemes. Table III compares the communication costs during exchange of messages in the proposed scheme ($AgroMobiBlock$) and other existing schemes. It can be observed that $AgroMobiBlock$ takes comparable communication costs to achieve more security and functionality features.

TABLE III
Comparison of Communication Costs

Protocol	No. of messages	Total cost (in bits)
Ali *et al.* [9]	5	5504
Chen *et al.* [10]	4	4960
Chae and Cho [11]	4	12896
Rangwani *et al.* [15]	5	4128
Wu and Tsai [16]	10	$1344 + 256n$
Vangala *et al.* [17]	6	4576
Bera *et al.* [18]	4	3456
Tian *et al.* [23]	2	$384s$ + 11712
Shuai *et al.* [24]	4	7616
Panda *et al.* [25]	5	$2N * 256$ + 4480
Eddine *et al.* [26]	4	2273
Fan *et al.* [27]	5	4480
Tomar & Tripathi [28]	3	4112
Itoo *et al.* [29]	4	1408
Jia *et al.* [30]	6	3616
AgroMobiBlock (Phase-1)	3	2848
AgroMobiBlock (Phase-2)	5	4608

Note: s: "number of pseudonyms of a drone in Tian *et al.*'s scheme" [23]; N: "a positive integer that specifies the number of public/private key pairs available to an IoT device in Panda *et al.*'s scheme" [25]; n: "number of agricultural equipment (sensor devices) in Wu and Tsai's scheme" [16].

B. Computation Costs Comparison

The proposed scheme is compared with the existing schemes in terms of the amount of cost expended in the computation of the operations involved in the scheme. We have used the broadly accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [61] to evaluate the execution time of various cryptographic primitives. The experiments for each cryptographic primitive are provided in the supplementary material.

The proposed $AgroMobiBlock$ scheme takes $6T_h + 4T_{ecm} + T_{eca}$ computation cost at the IoT smart node and the same cost $6T_h + 4T_{ecm} + T_{eca}$ at the mobile vehicle in the SNMV phase. It takes $24T_h + 8T_{ecm} + 2T_{eca} + 2T_{ECCDec}$ at both the mobile vehicle and fog server in the MVFS phase. Encryption and decryption do not lead to performance bottlenecks as they are only performed by mobile vehicles and fog servers, which are resource abundant. Encryption is only done once for each one-time registration of MV and FS. Table IV compares the computation costs during the exchange of messages in the proposed scheme ($AgroMobiBlock$) and other existing schemes. It can be observed that $AgroMobiBlock$ takes a comparable computation cost to achieve more security functionality features.

C. Comparison of Blockchain Features

This section gives a comparison of the level of involvement of blockchain in the existing blockchain-based schemes. It is observed from Table V that even though some schemes access the blockchain during authentication to store public keys or authentication messages, none of them used the blockchain to store secrets during registration and later access them during authentication, along with storing sensor data at the same time. Our proposed scheme makes maximum use of the single blockchain by using it in multiple phases and storing credentials along with sensor data.

D. Discussion on Performance Analysis

$AgroMobiBlock$ has lower communication cost compared to the schemes in [9], [10], [11], [16], [23], [24], and [25],

TABLE IV
COMPARISON OF COMPUTATION COSTS

Protocol	Smart device/Drone end	GSS/Server end
Ali *et al.* [9]	$11T_h$+T_{fe}+$3T_{senc/T_{sdec}}$ ≈ 5.738 ms	$8T_h + 5T_{senc/T_{sdec}}$ ≈ 0.445 ms
Chen *et al.* [10]	20 T_h ≈ 6.18 ms	17 T_h ≈ 0.935 ms
Chae and Cho [11]	8 T_{ecm} + 8 T_h + $2Teca$ ≈ 20.808 ms	–
Rangwani *et al.* [15]	8 T_h + 5 T_{ecm} ≈ 13.912 ms	7 T_h + T_{ecm} ≈ 1.059 ms
Wu and Tsai [16]	$2T_{bp}$ + $2T_{exp}$ + $2T_{enc/dec}$ + $1T_h$ + $1T_h$ ≈ 64.965 ms	$2T_{bp}$ + $2T_{exp}$ + $2T_{enc/dec}$ + $1T_h$ ≈ 9.407 ms
Vangala *et al.* [17]	$17T_h$ + $12T_{ecm}$ ≈ 32.645 ms	$9T_h$ + $4T_{ecm}$ ≈ 3.191 ms
Bera *et al.* [18]	$18T_h$ + $14T_{ecm}$ + $2T_{eca}$ + T_{poly} ≈ 39.758 ms	$7T_h$ + $6T_{ecm}$ + $2T_{eca}$ + T_{poly} ≈ 4.733 ms
Tian *et al.* [23]	$8T_{exp} + 9T_h$ ≈ 4.605 ms	–
Shuai *et al.* [24]	$13T_h$ + $3T_{exp}$ ≈ 4.701 ms	$7T_h$ + T_{exp} ≈ 0.457 ms
Panda *et al.* [25]	$(N+k+l+3)T_h + N*k*NT_h$ ≈ 110 ms	$4T_{SymmEnc}$ + $2T_{SymmDec}$ ≈ 10.915 ms
Eddine *et al.* [26]	$3T_h$ + $3T_{ecm}$ + T_{ECCEnc} + T_{ECCDec} + T_{CKDf} ≈ 19.279 ms	$2T_H$+ $3T_{ecm}$+ $2T_{ECCEnc}$+ T_{ECCDec}+ T_{CKDf} ≈ 6.858 ms
Fan *et al.* [27]	$4T_{bp}$ + $5T_h$ + $6T_{ecm}$ + $2T_{eca}$ + $3T_{SymmEnc}$ + $T_{SymmDec}$ ≈ 143.709 ms	$4T_{bp}$ + $5T_h$ + $6T_{ecm}$ + $2T_{eca}$ + $2T_{SymmEnc}$ + $T_{SymmDec}$ ≈ 22.738 ms
Tomar & Tripathi [28]	$4T_{ecm}$ + $7T_h$ ≈ 11.315 ms	$13T_{ecm}$ + $15T_h$ + $4T_{eca}$ ≈ 9.595 ms
Itoo *et al.* [29]	$5T_{ecm}$ + $3T_{eca}$ + $8T_h$ ≈ 13.96 ms	$7T_{ecm}$ + $5T_{eca}$ + $11T_h$ ≈ 5.333 ms
Jia *et al.* [30]	$2T_{ecm}$ + $13T_h$ ≈ 8.593 ms	$4T_{ecm}$ + $13T_h$ ≈ 3.411 ms
AgroMobiBlock (Phase 1)	$6T_h$+ $4T_{ecm}$+ T_{eca} ≈ 11.022 ms	$6T_H$+ $4T_{ecm}$+ T_{eca} ≈ 3.028 ms
AgroMobiBlock (Phase 2)	–	$24T_h$ + $8T_{ecm}$+ $2T_{eca}$ + $2T_{ECCDec}$ ≈ 8.068 ms

Note: k,l: "k^{th} and l^{th} devices communicate with each other such that $0 < k, l < N$ in Panda *et al.*'s scheme"; N: "a positive integer that specifies the number of public/private key pairs available to an IoT device in Panda *et al.*'s scheme". Consider $N = 20$, $k = 5$ and $l = 7$.

and significantly lower computation cost compared to [11], [15], [16], [25], [26], and [27]. Wu and Tsai's scheme [16] is a blockchain-based scheme, but it incurs a very high cost for computation and communication, lacks dynamic node addition, and is vulnerable to ESL and offline guessing attacks. Rangwani et al.'s scheme [15] has low computation cost, but it still lacks anonymity, untraceability, dynamic node addition and blockchain support, and cannot resist ESL and DoS attacks. *AgroMobiBlock* achieves all the security and functionality features with low communication and computation costs, which is not demonstrated in any of the compared schemes.

VIII. REAL-TIME PRACTICAL IMPLEMENTATION USING TESTBED EXPERIMENTATION

This section presents the environment of the real-time testbed experimental setup that implements the proposed model with both the authentication and key agreement between a sensor node and a mobile vehicle (SNMV Phase) and between a mobile vehicle and its respective fog server (MVFS Phase) as well as the registration phases as described in Section IV. The detailed description of the setup is given in the supplementary material. In addition, the experimental results for this implementation are also provided in the supplementary material.

IX. BLOCKCHAIN SIMULATION

As discussed in Section III, the simulations assume that the fog servers associated with an agricultural field form a P2P fog system that receive transactions from mobile vehicles. The P2P fog system creates either an AuthCred block or SensorData block to be added to the single hybrid blockchain. There are 11 fog servers assumed to be in a P2P fog system. One of the fog servers, elected as the leader/miner/proposer in a round-robin fashion, initiates the PBFT consensus algorithm [44]

TABLE V
COMPARISON OF BLOCKHAIN-RELATED FEATURES

Scheme	Purpose of blockchain	Usage of blockchain during registration	authentication	storing sensor data	storing secrets
Wu and Tsai *et al.*	*data packets distributed over multiple blockchains *dark web hides physical location of blockchain servers	×	×	✓	×
Vangala *et al.*	*store sensor data after authentication	×	×	✓	×
Bera *et al.*	*store sensor data after authentication	×	×	✓	
Eddine *et al.*	*store authentication results	×	×	×	×
Fan *et al.*	*store sensor data after authentication	×	×	✓	×
Tomar & Tripathi	*dual blockchain architecture with cloud and fog servers *multiple channels *stores public keys to extract identities	×	×	✓	×
Itoo *et al.*	*store authentication messages	×	✓	×	×
Jia *et al.*	*blockchain acts as PKI to store public keys and certificates	×	✓	✓	×
AgroMobiBlock	*store credentials during registration *access credentials during authentication *store sensor data on blockchain	✓	✓	✓	✓

for block creation, verification and inclusion into blockchain as discussed in detail in Algorithm 1.

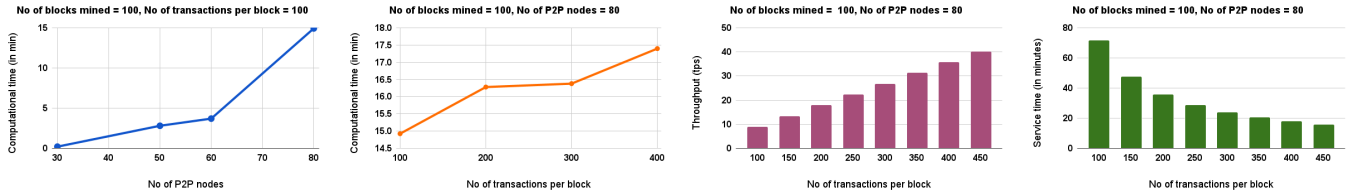
The blockchain simulations were performed on a server platform having the environment: "CentOS Stream 8, with 64 CPUs, each with 64-bit OS with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, 376 GiB RAM" using Node.js language with VS CODE 2019 [62]. The purpose of this simulation is to study the effect of an increase in the number of fog nodes, transactions, and blocks over the computational time.

In this simulation study, we have considered the synthetic data only. The IoT smart devices send the sensing data securely to their respective mobile vehicles and then the mobile vehicles also send the data securely to their nearby fog servers. The servers aggregate the genuine data as transactions, which are used to form blocks. Therefore, we do not consider any real-time agricultural-related data in this work.

The following scenarios are considered.

Scenario 1: Here, we measure the time to add a single block and monitor its variation with the network size, that is, the number of nodes in the network. We increase the network size from 30 to 80 server nodes. The graph in Fig. 7(a) depicts the relation between single block addition time, that is, consensus time, with the size of the network. The graph clearly shows that as the network size increases the single block addition time increases significantly. A PBFT-based voting mechanism requires more communication cost of $O(C^{N_f})$ where C is the number of messages exchanged in the pre-prepare, prepare and commit phases executed by the N_f servers participating in the consensus [44]. More number of nodes leads to increased time to reach consensus. This cost is in addition to the communication cost computed for Algorithm 1 in Section IV-E.

Scenario 2: Here, we measure the time to add a single block and monitor its variation with the number of transactions in a block. The network size is fixed to 80 server nodes and the blockchain length is fixed at 100 blocks. The transactions count is varied from 100 to 400 transactions per block. In the graph in Fig. 7(b), it is clearly observed that an increase in the number of transactions leads to an increase in the



(a) (b) (c) (d)

Fig. 7. Blockchain simulation results: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4.

consensus computational time. This is because as the number of transactions in a block increase, each node in the P2P network has to verify the signature, block hash and Merkle tree root for the increased number of transactions adding to the overall time taken by the consensus.

Scenario 3: This scenario measures the throughput of the blockchain as the number of transactions processed per second. This is obtained as the ratio of the number of transactions per block to the average block time taken in seconds. Fig. 7 (c) depicts the throughput of the blockchain with the increase in the number of transactions per block from 100 to 450. It is observed that the throughput increases when the number of transactions per block is increased while the number of P2P nodes and the number of blocks mined are fixed at 80 and 100, respectively. As the number of transactions per block increases, the time taken to process each block also increases, leading to higher number of transactions being processed per second over the entire network.

Scenario 4: Here, the service time of a node in the blockchain is the time that a node is busy in processing the transactions of the block it received. A node places its transactions in a memory pool. If a transaction of a block being mined is unavailable in the memory pool, it requests the network to obtain it. To obtain the service time of a mining node, the ratio of the number of mining nodes to the time to mine each block is divided by the number of transactions per block. Fig. 7 (d) shows that the service time of the node decreases as the number of transactions per block decreases.

In summary, a single block addition time, that is, consensus time is dependent on the size of the P2P network, but not on block size or blockchain length. Power consumption of Algorithm 1 is directly proportional to the algorithm execution time. Therefore, the blockchain simulation results show that the execution time also directly reflect the power consumption of the algorithm.

X. Conclusion

We designed a robust authenticated key agreement scheme using a hybrid blockchain with the help of mobile vehicles for a precision agricultural IoT network. The active blockchain is accessed to extract credentials for key agreement between a mobile vehicle and a fog server. Fog servers securely aggregate sensor data that form blocks of transactions. Blocks are mined with the help of a voting-based PBFT consensus mechanism. Detailed security analysis and comparative study reveal that the proposed scheme resists various attacks, offers more functionality attributes, and has comparable communication and computational costs to other competing schemes. Finally, the real-time testbed experiments are performed to exhibit the proposed scheme's practical usage.

Acknowledgment

The authors would like thank the anonymous reviewers and the Associate Editor for providing constructive and generous feedback.

References

[1] United Nations, Department of Economic and Social Affairs, Population Division. (2019). *World Population Prospects*. [Online]. Available: https://population.un.org/wpp/

[2] R. Ramakumar, "Agriculture and the COVID-19 pandemic: An analysis with special reference to India," *Rev. Agrarian Stud.*, vol. 10, no. 1, pp. 1–40, 2020.

[3] *Economic Crisis in Sri Lanka*. Accessed: Mar. 2022. [Online]. Available: https://www.thehindu.com/business/explained-what-caused-the-sri-lankan-economic-crisis/article36314148.ece

[4] *Decade's Worst Food Shortage in North Korea*. Accessed: Mar. 2022. [Online]. Available: https://timesofindia.indiatimes.com/world/rest-of-world/north-korea-says-its-facing-worst-food-shortage-in-decade/articleshow/84404761.cms

[5] "The war in Ukraine is exposing gaps in the world's food-systems research," *Nature*, vol. 604, no. 7905, pp. 217–218, 2022. [Online]. Available: https://www.nature.com/articles/d41586-022-00994-8

[6] C. Lu. (2022). *Russia's Invasion Unleashes 'Perfect Storm' in Global Agriculture*. [Online]. Available: https://foreignpolicy.com/2022/03/24/russia-war-ukraine-food-crisis-wheat-fertilizer/

[7] M. Torky and A. E. Hassanein, "Integrating blockchain and the Internet of Things in precision agriculture: Analysis, opportunities, and challenges," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105476.

[8] Y.-P. Lin et al., "Blockchain: The evolutionary next step for ICT E-agriculture," *Environments*, vol. 4, no. 3, p. 50, Jul. 2017.

[9] R. Ali, A. Pal, S. Kumari, M. Karuppiah, and M. Conti, "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Gener. Comput. Syst.*, vol. 84, pp. 200–215, Jul. 2018.

[10] M. Chen, T.-F. Lee, and J.-I. Pan, "An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring," *Sensors*, vol. 19, no. 5, p. 1146, 2019.

[11] C.-J. Chae and H.-J. Cho, "Enhanced secure device authentication algorithm in P2P-based smart farm system," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 6, pp. 1230–1239, Nov. 2018.

[12] S. Alyahya, W. U. Khan, S. Ahmed, S. N. K. Marwat, and S. Habib, "Cyber secure framework for smart agriculture: Robust and tamper-resistant authentication scheme for IoT devices," *Electronics*, vol. 11, no. 6, p. 963, Mar. 2022.

[13] A. Bothe, J. Bauer, and N. Aschenbruck, "RFID-assisted continuous user authentication for IoT-based smart farming," in *Proc. IEEE Int. Conf. RFID Technol. Appl. (RFID-TA)*, Pisa, Italy, Sep. 2019, pp. 505–510.

[14] J. Arshad et al., "A novel remote user authentication scheme by using private blockchain-based secure access control for agriculture monitoring," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Lahore, Pakistan, Feb. 2020, pp. 1–9.

[15] D. Rangwani, D. Sadhukhan, S. Ray, M. K. Khan, and M. Dasgupta, "An improved privacy preserving remote user authentication scheme for agricultural wireless sensor network," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 3, Mar. 2021, Art. no. e4218.

[16] H.-T. Wu and C.-W. Tsai, "An intelligent agriculture network security system based on private blockchains," *J. Commun. Netw.*, vol. 21, no. 5, pp. 503–508, Oct. 2019.

[17] A. Vangala, A. K. Sutrala, A. K. Das, and M. Jo, "Smart contract-based blockchain-envisioned authentication scheme for smart farming," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10792–10806, Jul. 2021.

[18] B. Bera, A. Vangala, A. K. Das, P. Lorenz, and M. K. Khan, "Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment," *Comput. Standards Interfaces*, vol. 80, Mar. 2022, Art. no. 103567.

[19] A. Mitra, B. Bera, A. K. Das, S. S. Jamal, and I. You, "Impact on blockchain-based AI/ML-enabled big data analytics for cognitive Internet of Things environment," *Comput. Commun.*, vol. 197, pp. 173–185, Jan. 2023.

[20] Z. Shelby, K. Hartke, and C. Bormann. (2014). *The Constrained Application Protocol (CoAP)*. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7252

[21] A. Vangala, A. K. Das, N. Kumar, and M. Alazab, "Smart secure sensing for IoT-based agriculture: Blockchain perspective," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17591–17607, Aug. 2021.

[22] A. Vangala, A. K. Das, V. Chamola, V. Korotaev, and J. J. P. C. Rodrigues, "Security in IoT-enabled smart agriculture: Architecture, security solutions and challenges," *Cluster Comput.*, pp. 1–24, Apr. 2022, doi: 10.1007/s10586-022-03566-7.

[23] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102354.

[24] M. Shuai, L. Xiong, C. Wang, and N. Yu, "A secure authentication scheme with forward secrecy for industrial Internet of Things using Rabin cryptosystem," *Comput. Commun.*, vol. 160, pp. 215–227, 2020.

[25] S. S. Panda, D. Jena, B. K. Mohanta, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Authentication and key management in distributed IoT using blockchain technology," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12947–12954, Aug. 2021.

[26] M. S. Eddine, M. A. Ferrag, O. Friha, and L. Maglaras, "EASBF: An efficient authentication scheme over blockchain for fog computing-enabled Internet of Vehicles," *J. Inf. Secur. Appl.*, vol. 59, Jun. 2021, Art. no. 102802.

[27] Q. Fan, J. Chen, L. J. Deborah, and M. Luo, "A secure and efficient authentication and data sharing scheme for Internet of Things based on blockchain," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102112.

[28] A. Tomar and S. Tripathi, "Blockchain-assisted authentication and key agreement scheme for fog-based smart grid," *Cluster Comput.*, vol. 25, no. 1, pp. 451–468, Feb. 2022.

[29] S. Itoo, A. A. Khan, V. Kumar, A. Alkhayyat, M. Ahmad, and J. Srinivas, "CKMIB: Construction of key agreement protocol for cloud medical infrastructure using blockchain," *IEEE Access*, vol. 10, pp. 67787–67801, 2022.

[30] X. Jia, M. Luo, H. Wang, J. Shen, and D. He, "A blockchain-assisted privacy-aware authentication scheme for Internet of Medical Things," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21838–21850, Nov. 2022.

[31] D. Mishra, D. Dharminder, P. Yadav, Y. S. Rao, P. Vijayakumar, and N. Kumar, "A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102648.

[32] N. Andola, S. Prakash, R. Gahlot, S. Venkatesan, and S. Verma, "An enhanced smart card and dynamic ID based remote multi-server user authentication scheme," *Cluster Comput.*, vol. 25, pp. 3699–3717, May 2022.

[33] S. Qiu, G. Xu, H. Ahmad, G. Xu, X. Qiu, and H. Xu, "An improved lightweight two-factor authentication and key agreement protocol with dynamic identity based on elliptic curve cryptography," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 2, pp. 978–1002, 2019.

[34] J. Liu, R. Liu, and Y. Lai, "Risk-based dynamic identity authentication method based on the UCON model," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, Mar. 2022.

[35] Y. Li, Q. Cheng, and W. Shi, "Security analysis of a lightweight identity-based two-party authenticated key agreement protocol for IIoT environments," *Secur. Commun. Netw.*, vol. 2021, pp. 1–6, Feb. 2021.

[36] D. S. Gupta, S. H. Islam, M. S. Obaidat, P. Vijayakumar, N. Kumar, and Y. Park, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments," *IEEE Syst. J.*, vol. 15, no. 2, pp. 1732–1741, Jun. 2021.

[37] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[38] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[39] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[40] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Proc. 16th Annu. Int. Cryptol. Conf. (CRYPTO)*, Barbara, CA, USA, 1996, pp. 104–113.

[41] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 2, pp. 391–406, Mar. 2020.

[42] E. Bertino, N. Shang, and S. S. Wagstaff Jr., "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE Trans. Depend. Sec. Comput.*, vol. 5, no. 2, pp. 65–70, Apr./Jun. 2008.

[43] D. Basin, C. Cremers, and S. Meier, "Provably repairing the ISO/IEC 9798 standard for entity authentication," *J. Comput. Secur.*, vol. 21, no. 6, pp. 817–846, 2013.

[44] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[45] H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, Aug. 2019.

[46] E. Barker, "Recommendation for key management," NIST, Gaithersburg, MD, USA, Tech. Rep. SP 800-57 Part 1 Rev. 4, Jan. 2016. Accessed: May 2021. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

[47] J. Hu, M. J. Reed, M. Al-Naday, and N. Thomos, "Hybrid blockchain for IoT—Energy analysis and reward plan," *Sensors*, vol. 21, no. 1, pp. 1–21, 2021.

[48] C. Smith. *B Merkle Tree*. Accessed: Apr. 2022. [Online]. Available: https://medium.com/proxima-one/b-merkle-tree-824952837bfe

[49] C. Smith and A. Rusnak, "Dynamic Merkle B-tree with efficient proofs," 2020, *arXiv:2006.01994*.

[50] J. Tu, J. Zhang, S. Chen, T. Weise, and L. Zou, "An improved retrieval method for multi-transaction mode consortium blockchain," *Electronics*, vol. 9, no. 2, p. 296, Feb. 2020.

[51] H. Feng, J. Wang, and Y. Li, "An efficient blockchain transaction retrieval system," *Future Internet*, vol. 14, no. 9, p. 267, Sep. 2022.

[52] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC)*, in Lecture Notes in Computer Science, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.

[53] Y. Lin, X. Wang, Q. Gan, and M. Yao, "A secure cross-domain authentication scheme with perfect forward security and complete anonymity in fog computing," *J. Inf. Secur. Appl.*, vol. 63, Dec. 2021, Art. no. 103022.

[54] AVISPA. (2020). *Automated Validation of Internet Security Protocols and Applications*. Accessed: May 2021. [Online]. Available: http://www.avispa-project.org/

[55] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000*, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, 2000, pp. 139–155.

[56] L. Wu, J. Wang, K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 319–330, Feb. 2019.

[57] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, 1999, pp. 388–397.

[58] M. H. Au, K. Liang, J. K. Liu, R. Lu, and J. Ning, "Privacy-preserving personal data operation on mobile cloud—Chances and challenges over advanced persistent threat," *Future Gener. Comput. Syst.*, vol. 79, pp. 337–349, Feb. 2018.

[59] F. J. Abdullayeva, "Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm," *Array*, vol. 10, Jul. 2021, Art. no. 100067.

[60] L.-X. Yang, K. Huang, X. Yang, Y. Zhang, Y. Xiang, and Y. Y. Tang, "Defense against advanced persistent threat through data backup and recovery," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2001–2013, Jul. 2021.

[61] (2020). *MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library*. Accessed: May 2021. [Online]. Available: https://github.com/miracl/MIRACL

[62] K. Khullar. (2019). *Implementing PBFT in Blockchain*. Accessed: Jul. 2020. [Online]. Available: https://medium.com/coinmonks/implementing-pbft-in-blockchain-12368c6c9548