

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Redactable Blockchain-Assisted Secure Data Aggregation Scheme for Fog-Enabled Internet-of-Farming-Things

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Mishra, R., Ramesh, D., Bellavista, P., Edla, D.R. (2023). Redactable Blockchain-Assisted Secure Data Aggregation Scheme for Fog-Enabled Internet-of-Farming-Things. IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, 20(4), 4652-4667 [10.1109/TNSM.2023.3322442].

Availability:

This version is available at: <https://hdl.handle.net/11585/952084> since: 2024-01-04

Published:

DOI: <http://doi.org/10.1109/TNSM.2023.3322442>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

R. Mishra, D. Ramesh, P. Bellavista and D. R. Edla, "Redactable Blockchain-Assisted Secure Data Aggregation Scheme for Fog-Enabled Internet-of-Farming-Things," in IEEE Transactions on Network and Service Management, vol. 20, no. 4, pp. 4652-4667, Dec. 2023

The final published version is available online at:

<https://dx.doi.org/10.1109/TITS.2023.3333128>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)





When citing, please refer to the published version.



Redactable Blockchain-Assisted Secure Data Aggregation Scheme for Fog-enabled Internet-of-Farming-Things

Journal:	<i>IEEE Transactions on Network and Service Management</i>
Manuscript ID	TNSM-2022-05531
Manuscript Type:	Regular Issue submission
Date Submitted by the Author:	03-Aug-2022
Complete List of Authors:	Mishra, Rahul; Indian Institute of Technology, Computer Science and Engineering Ramesh, Dharavath ; Indian Institute of Technology, Computer Science and Engineering Bellavista, Paolo; University of Bologna, Computer Science and Engineering Edla, Damodary Reddy; National Institute of Technology Goa, Computer Science and Engineering
Keywords:	Internet-of-Farming-Things (IoFT), Fog computing, Data aggregation, Redactable blockchain

Redactable Blockchain-Assisted Secure Data Aggregation Scheme for Fog-enabled Internet-of-Farming-Things

Rahul Mishra , Student Member, IEEE, Dharavath Ramesh , Senior Member, IEEE, Paolo Bellavista , Senior Member, IEEE, and Damodar Reddy Edla , Senior Member, IEEE

Abstract—Internet-of-Farming Things (IoFT)-enabled smart agriculture can collect data more reliably and frequently to track the crop's status and other significant information. Considering that smart agriculture requires working with substantial amounts of sensitive data. In light of this, frequent data processing may threaten the confidentiality and integrity of data and IoFT device privacy. Although numerous privacy-preserving data aggregation methods have been put out to address these issues, they also have certain security vulnerabilities, such as inadequate data confidentiality, collusion attack, and malicious data mining attacks. Therefore, we introduce a three-tier architecture-assisted redactable blockchain-based secure data aggregation method with source authentication for the fog-enabled IoFT. This work provides an efficient and secure two-level data aggregation model. The proposed model supports resistance to collusion and malicious data mining threats launched by internal or external attackers. It can also achieve perfect data confidentiality and integrity against a malicious aggregator and an inquisitive control center for an authorized IoFT device. Specifically, the detailed performance analysis and theoretical concrete security proofs demonstrate the practicability and efficiency of the proposed model.

Keywords—Internet-of-Farming-Things (IoFT), Fog computing, Data aggregation, Redactable blockchain.

I. INTRODUCTION

AGRICULTURE is the global primary sector of the economy, which contributes significantly to economic growth and sustainability [1]. Presently, due to expanding global population, people want sustainably grown grains. Smart agriculture can meet the requirements, which can also assist the farmers in taking advantage of potential opportunities. It brings a new way of agriculture production. It enhances the agriculture information perception, intelligent control, quantitative decision-making, and quality service through a deep integration of fog-computing and Internet-of-Farming-Things (IoFT) [2], [3]. IoFT is introduced to automate the farming process, such as collecting crop-related information like humidity level,

soil's P^H - value, soil's water level, nutrient-level of soil, etc. IoFT sends all the collected crop-status data to cloud-assisted Agriculture Control Center (ACC). At ACC, this real-time data is used to take action for irrigation, fertilizer spreading, pest management, and any other related activities for the significant growth of crops. However, this cloud-assisted framework has its own set of issues regarding data confidentiality and integrity, high latency, etc. [4], [5].

Fog computing has recently gained popularity as an efficient solution that can efficiently manage data aggregation tasks while transferring the data to remote cloud servers. Taking Fog-computing assisted smart agriculture as an example, a fog server can aggregate the crop-status data collected by IoFT devices; then transfer it to ACC. Through its control method, ACC can successfully analyze the collected data to take suitable action for the proper growth of crops. However, due to the limited resources of fog-servers, the existing data-security mitigation is not fully applicable to Fog-computing architecture. Some groups of researchers [6], [7] have highlighted that it is an essential research topic to realize security threats, i.e., data integrity and confidentiality, false data-injection attacks, privacy-preserving, and high-performance interactions. Also, a secure data aggregation in fog-enabled IoFT framework is still considered as a work in progress scenario. In short, secure data aggregation have a significant role in balancing IoFT devices' privacy-preserving and data integrity and confidentiality; also one of the essential aspects of the development of smart agriculture.

Fortunately, the advent of blockchain has made it possible to address the security issues of the data aggregation models [8], [9]. Blockchain technology integrates the distributed ledger, smart contract, and consensus algorithm to ensure immutable data storage. Nevertheless, several security issues with blockchain are starting to surface due to the exponential expansion of IoFT and fog-device performance. Generally, a group of malicious fog-server can launch powerful 51% attack to record the incorrect data on the blockchain, which will cause disastrous effects [10]. In this situation, Ateniese et al. [11] introduced redactable blockchain architecture, which allows for editing in any previously created block over the blockchain network. A redactable blockchain architecture can easily edit the incorrect data, also ensuring the security of the data. Specifically, in IoFT paradigm, for some instances, any IoFT device becomes unstable and may upload inaccurate aggregate data to ACC via Fog-layer and blockchain. Then, ACC will

The first author Rahul Mishra and the second author Dharavath Ramesh are with the Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad 826004, India. E-mail: rahul.18dr0107@cse.iitism.ac.in and drramesh@iitism.ac.in. The third author is with the with the Department of Computer Science and Engineering, University of Bologna, 40136 Bologna, Italy. E-mail: paolo.bellavista@unibo.it. The Fourth author Damodar Reddy Edla is with the Department of Computer Science and Engineering, National Institute of Technology, Farmagudi, Goa-403401, India, E-mail: dr.reddy@nitgoa.ac.in.

¹A preliminary version of this paper was presented at the IoT-PRO workshop with 20th IEEE PerCom - 2022 [14].

also take the wrong analysis for crop-status, which may result in catastrophic consequences. So, in such a scenario, the redactability of the recorded block can provide an efficient way to edit or delete the respective block over the blockchain at Fog-layer.

Therefore, in our preliminary conference version of this paper [12], we address that the key challenges in the data aggregation model have privacy-preserving, data integrity, and confidentiality brought on by malicious activities of cloud servers and fog-servers. However, as we will go into considerable detail in this paper, it is also vital to deal with various significant security threats, i.e., fake data injection attacks and tempering attacks in fog-enabled data aggregation models. Accordingly, we also considered the redactable blockchain enabled secure data aggregation model in the Fog paradigm to provide required editing against any wrong data collection over the blockchain.

In this article, we extend our work [12] for secure layered data aggregation model to Fog-enabled smart agriculture paradigm with IoFT devices to provide more efficient and computationally infeasible data-aggregation. This extended work considers the computationally infeasible efficient paillier cryptosystem [13] and aggregate certificateless signature scheme to support security against malicious data mining attack by internal and external attackers and false data injection attack. Moreover, we also consider the instantly redactable blockchain framework for extended work to provide efficient editing operations over the blockchain network [14], where a random committee with significant honest fraction is selected first and then committee-members would initiate the redaction. In the proposed model, IoFT devices collect the crop-status data and send the encrypted data to the local fog-server at the fog layer, which performs local aggregation. Afterward, the defined leader of the fog-server creates the global aggregation and uploads it to the blockchain. Then, by accessing the data from the blockchain, ACC can analyze this data in real-time for better growth of crops. In such conditions, we assure the proper privacy for IoFT devices, and that only ACC can obtain the aggregated data with adequate data integrity.

A. Our contributions

The main contributions of the proposed model are summarized as follows;

- 1) We present a three-tier architecture-based data aggregation model, which provides a secure infrastructure that efficiently uses local resources - IoFT devices; significantly contributes to achieving secure data aggregation in fog-enabled smart agriculture paradigm.
- 2) We provide a computationally intractable data aggregation model with efficient IND-CCA2 secure paillier cryptosystem and aggregate certificateless signature scheme. It can simultaneously provide proper computational infeasibility against data mining attacks with proper source authentication and data integrity without using any extra cryptographic primitives.
- 3) We also consider efficient instantly redactable blockchain framework into fog-layer to enhance redactability with

security and flexibility of fog-layer against collusion attack.

- 4) Moreover, the detailed theoretical security analysis illustrates that the proposed data aggregation model can meet all the standard security properties; while keeping high efficiency.
- 5) Finally, the detailed performance analysis with significant experiments shows the practicability and feasibility of the proposed model in terms of computational and communication overheads.

B. Organization

The rest of the paper is organized as follows; Section II introduces some previously designed data aggregation models. In Section III, we describe the system model, the adversary model, and some other related definitions with design objectives. Section IV includes the basics preliminaries for the proposed model. In Section V, we discuss the complete demonstration of the proposed model. Detailed security analysis of the proposed model is described in Section VI. In Section VII, we analyze the performance of the proposed model in terms of communication and computation overheads. Finally, section VIII concludes this proposed model.

II. LITERATURE REVIEW

This section briefly overviews the existing models to design secure data aggregation schemes. This section is classified as traditional fog-enabled data aggregation models and blockchain-assisted fog-enabled data aggregation models.

A. Traditional fog-enabled data aggregation models

Bonomi et al. [15] were the first to introduce the concept of Fog computing. They presented an overview of fog and introduced some significant Fog computing based applications. Roman et al. [16] introduced the data aggregation issue in Fog computing regarding privacy and security. In 2017, Lu et al. [17] introduced a lightweight data aggregation model for fog-enabled paradigm with paillier homomorphic encryption and Chinese Remainder Theorem (CRT) to support secure aggregation and overlay fake data at the initial stage on fog-layer. Further, they established a method to sensor low-pass external data injection attacks. One year later, Wang et al. [18] proposed a Castagnos-Laguillaumine cryptosystem and short signature based secure data aggregation model in Fog computing. In this model, edge devices aggregate the collected data and send it to cloud storage with high communication overhead. Simultaneously, Lyu et al. [19] proposed a fault-tolerant differential private aggregate model with additive homomorphic encryption. The Gaussian distribution is used to ensure the privacy of data. This model is vulnerable to false data injection attacks, due to an additional round of communication during the failure of any IoT devices. This model also suffered from extra communication overhead. In 2020, Saleem et al. [20] introduced a paillier homomorphic encryption-based privacy-preserving data aggregation model, ensuring security against false data injection and reply attacks.

Nevertheless, if any IoT devices breakdown and aggregation is delayed, this model incurs high communication overheads. In 2021, Mohammadali et al. [21] compiled an efficient data aggregation model, which provides a multidimensional data aggregation model with a signature-based authentication model. It supports row and column level data aggregation to provide more precise analysis at Cloud Service Provider (CSP). Moreover, it also supports a batch verification model with a high computational overhead.

Liu et al. [22] introduced a certificateless data aggregation model for IoT enabled smart grid. This model ensures resistant against reply, impersonation, and false data injection attacks. However, this model fails to facilitate fault tolerance and has a high computational overhead. In 2021, Wang et al. [23] also proposed a secure privacy-preserving aggregation model for IoT assisted smart grid. However, this model also ignored the identity of users at the smart meter level. Simultaneously, Khan et al. [24] proposed a secure data aggregation model for a smart grid with Boneh-Gon-Nisson (BGN) cryptosystem and Elliptic curve digital signature scheme (ECDSA) based authentication to resist reply and false data injection attacks. Moreover, Song et al. [25] proposed a new flexible privacy-preserving data aggregation model for smart agriculture. This model is designed to collect crop-status data with flexible security property of Elgamal-cryptosystem with detailed security proofs. This model also offers optional data aggregation in the virtual aggregation area. However, this model fails to support resistance against false data injection attack and malicious data mining attack. Recently, Wang et al. [26] introduced a lightweight non-interactive privacy-preserving data aggregation method for constrained devices. This model employs a trusted execution paradigm to avoid the necessity for trusted entities and reduce overheads.

B. Blockchain-assisted fog-enabled data aggregation models

The emergence of blockchain technology introduces creative solutions for resolving trust issues; the blockchain is tamper-proof and decentralized [27], [28]. Guan et al. [29] introduced a blockchain based model for secure data aggregation model of smart grid. In 2019, Liang et al. [30] designed a blockchain based distribution protection architecture to strengthen modern power systems' privacy-preserving functionalities against security attacks. A work compiled by Siguang et al. [31], introduced anonymous data aggregation and a double blockchain-assisted secure model for a fog-enabled smart grid. This research introduced a three-tier architecture based data aggregation framework using blockchain and Fog computing. This model enables robust support for secure and efficient data aggregation in the smart grid paradigm. In 2021, Niu et al. [32] introduced a methodology for operating aggregate statistics over private correlated data based on relying on differential privacy. This model also enhances the utility of aggregated data and ensures arbitrage free-ness. Simultaneously, Yan et al. [33] compiled a blockchain-assisted provable, reliable, and privacy-preserving data aggregation model with paillier cryptosystem. However, this model suffers from collusion attacks and false data injection attacks.

Simultaneously, Lu et al. [34] compiled a lightweight edge blockchain-assisted data aggregation model for smart grid data security and privacy-preserving. This model efficiently attains resistivity against false data injection attacks and enhances the systems' robustness. However, this model did not consider the malicious data mining attack from internal and external adversaries and collusion attacks. Recently, Verma et al. [35] proposed a computationally secure data aggregation model in pairing-free certificate-based signature architecture. This model introduces an efficient data aggregation model. However, it also does not consider collusion attacks. Moreover, a systematic survey for secure data aggregation models in fog environment can be found in [36], [37].

III. SYSTEM MODEL, ADVERSARY MODEL, AND DESIGN GOALS

In this section, we illustrate the system model and adversary model with design objectives of the proposed model.

A. System model

The system model of the proposed data aggregation model consists of three layer - IoFT layer, Fog layer, and Cloud layer. Fig. 1 illustrates all the entities involved in the system model.

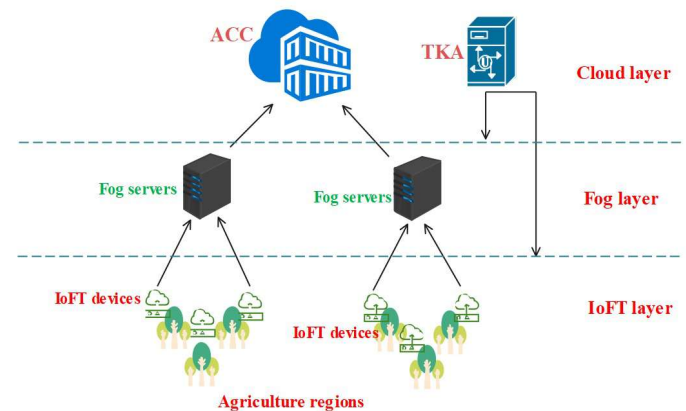


Fig. 1: System model of proposed data aggregation model

1) *IoFT layer*: This layer includes number of IoFT devices with further division of n - regions of ACC. Afterward, we assume that there are ϑ_i agriculture fields in region - i , and each agriculture field deploy IoFT devices to monitor the crop-status and important information. We represent FS_i as fog server in i^{th} - region and $IT_{i,j}$ - j^{th} IoFT device in region i , i.e., $j \leq \vartheta_i$. $IT_{i,j}$ will collect the crop-status data and send it to ACC via fog servers (FS_i).

2) *Fog layer*: The fog layer divides the ACC area into n - regions and the data for each region is locally aggregated by the corresponding fog servers. To attain the proper data security in the fog layer, we employ the efficient paillier cryptosystem and certificateless signature scheme. Moreover, we select a group of fog servers as authorized nodes for fog layer blockchain architecture to complete the block generation procedure over the blockchain. We employ committee-endorsing

mechanism with verifiable random function (VRF) to select authorized nodes among all fog servers (FS_i) [38]. The VRF is utilized to assure the random solution of authorized nodes among all the fog servers. Moreover, we also consider a leader-node with block right-ownership. Note that the leader-node is randomly selected among all the authorized nodes. Initially, the leader-node collects the transactions from all fog servers, packs it into a block, and finally broadcasts the block to other authorized nodes. Then, these authorized nodes validate it and upload it to the blockchain. Also, the authorized nodes and leader-node perform the redactability procedure over the fog layer blockchain.

3) *Cloud server*: This layer consists ACC and Trusted Key-Authority (TKA). ACC can collect all the aggregated data from fog servers (FS_i) and perform real-time analysis. Note that ACC reads the data every η minute to make it easier to monitor crop status at a certain time. Moreover, TKA is a trusted authority that sets up the complete system parameter and distributes it to other involved entities. TKA will be offline after completing the required tasks.

B. Adversary model

In the adversary model, initially, we consider TKA as a fully trustworthy entity that any adversaries or other entities cannot compromise through a secure channel. However, IoFT devices, fog servers, and ACC are honest but curious, which will honestly follow the defined procedure of the proposed model but may launch severe attacks, i.e., collusion, modification, data mining attacks, etc. So, to demonstrate the potential security threats in the proposed model, we consider probabilistic polynomial time (PPT) adversaries as follows;

- 1) A PPT adversary (A_I) may launch attack to break the confidentiality and privacy-preserving of the proposed model with malicious fog servers.
- 2) An another PPT adversary (A_{II}) may intercept the communication channel between IoFT layer-to-Fog layer and Fog layer-to-ACC to perform malicious data mining attack to temper the aggregated data, where A_{II} can block the communication between layers and compromise the ACC to obtain useful information from decrypted data.
- 3) Moreover, A_{II} may also launch a false data injection attack to fog servers to pass the authentication and try to access the globally aggregated data at fog layer.
- 4) Moreover, malicious ACC may collude with malicious fog servers and incentivized some authorized nodes to launch collusion and modification attacks to IoFT's aggregated data.

C. Design goals

Based on the system model and adversary model with possible attacks, the proposed model should achieve the following objectives;

- 1) **Confidentiality and Privacy-preserving** - Data aggregation should adhere to proper confidentiality such that even if an adversary launches eavesdropping on the communication channel, such adversary cannot have access

to confidential aggregated data. Moreover, the proposed model also guarantees the privacy-preserving; even if a false data injection attack is successful, none of the adversaries can access information about IoFT devices.

- 2) **Integrity and Authentication** - The proposed model should ensure the proper integrity of the aggregated data against any modification attack. Also, the proposed model provides the authenticity of all IoFT devices, fog servers during data aggregation and transmission.
- 3) **Efficiency and accuracy**- Taking into account the system participants' capacities and the frequency of data processing, the proposed data aggregation model should be efficient with computational and communication overheads in terms of off-chain and on-chain operations. Moreover, ACC should be able to ensure the accuracy of the final aggregated results it recovers; otherwise, ACC might take the incorrect action for the crop's status.

IV. PRELIMINARIES

This section describes some preliminaries to support the proposed scheme. We also give a brief explanation of useful computational assumptions.

A. Efficient Paillier cryptosystem

Specifically, the efficient paillier cryptosystem [13] consists of three algorithms such as Key-Gen(), Encryption(), and Decryption(). These are described in the following manner.

- 1) **Key-Gen** () - given security parameter $k > 1$, randomly select two large prime p, q of same length, i.e, $N = pq$ relative to which the Decisional Composite Residuosity Assumption (DCPR) is hard in $Z_{N^2}^*$ and τ , where $\tau \ll |N|$ and also select $g \in Z_{N^2}^*$, $H : \{0, 1\}^* \rightarrow Z_N$. Finally, set $\lambda = lcm(p-1, q-1)$ as private-key and (g, H, N) as public-key.
- 2) **Encryption** () - choose random number $r \in_R \{0, 1\}^\tau$ and set $z = [H(m||r)]^N \pmod{N^2}$, and $Ms = (m||r)$ for given message m , generate ciphertext as $c = [g^M \cdot z \pmod{N^2}]$
- 3) **Decryption** () - given ciphertext c and private key λ , compute $M = \left[\frac{L(c^\lambda \pmod{N^2})}{L(g^\lambda \pmod{N^2})} \right] \pmod{N}$, and parse $M = m||r$, then compute $z = g^{-r} \cdot c \pmod{N^2}$. If $z \stackrel{?}{=} [H(m||r)]^N \pmod{N^2}$, return decrypted actual message m otherwise return invalid.

B. Computational assumptions

Consider a multiplicative cyclic group $G_1 = \langle g_0 \rangle$ of prime order p with generator g_0 , where p is a large prime number, i.e., $p = 2q + 1$, q is also a large prime number.

1) *Discrete-Logarithm Problem (DL-Problem)*[39]: for given $d \in G_1$, calculate $y \in Z_p$ as $d = g_0^y$.

2) *Computational Diffie-Hellman Problem (CDH-Problem)*[39]: for randomly selected $a, b \in Z_p^*$, given tuple $\langle g_0, g_0^a, g_0^b \rangle$, compute the value of g_0^{ab} .

Moreover, the notations used in this work are summarized in Table II.

TABLE I: Symbols table

Notations	Definitions
$k > 1$	Security parameter
G_1	The multiplicative cyclic group with generator g_0
(N, g)	public-key of efficient paillier cryptosystem
(λ, κ)	private-key of efficient paillier cryptosystem
ms_{ij}	crop-status data
$(ID_{IT_{ij}}, ID_{FS_i})$	Identities of IoFT devices and fog servers, respectively
H_1, H_2	Collision resistant cryptographic function
$\vec{a} = (a_1, a_2, \dots, a_n)$	super linear sequence
$(ms_{priv_{IT_{ij}}}, ms_{pub_{IT_{ij}}})$	master-signing-key-pairs for IoFT devices
$(ms_{priv_{FS_i}}, ms_{pub_{FS_i}})$	master-signing-key-pairs for fog servers
C_i	Aggregation result of crop-status data in region of i^{th} - fog sever
C_{AS}	Aggregation result of crop-status data for all IoFT devices
σ_i	tag-value for locally aggregated data by fog servers
σ_{AS}	Aggregated tag-value for locally aggregated data by fog servers
$(MK_{IT_{ij}}, PK_{IT_{ij}})$	signing-key-pairs of IoFT devices
(MK_{FS_i}, PK_{FS_i})	signing-key-pairs for fog servers
$D = (CID_{IT_{ij}}, \sigma_{ij}, T_{mt})$	Collected crop-status data

V. CONSTRUCTION OF THE MODEL

This section describes the detailed construction of the proposed secure blockchain-based data aggregation scheme for IoFT-enabled smart agriculture. Specifically, this section is divided into five subsections - System-initialization phase, Registration phase for all involved entities (IoFT devices, fog-servers, ACC), IoFT-layer data generation phase, Fog-layer data aggregation phase, and Data Analysis at ACC phase. In the first phase, system initialization is mainly responsible for generating the required system parameters for all involved entities. In the registration phase, all the IoFT devices and fog servers registered themselves over the network. Also, the three-layer architecture of the proposed data aggregation model is following the data generation phase, data aggregation phase, and data analysis phase. Initially, IoFT devices collect the crop-status data and send the encrypted version to fog servers; then, fog servers aggregate it and send the aggregated version to the blockchain. Afterward, ACC accesses the blockchain to get the aggregated crop-status data and analyze it by decrypting it with corresponding keys. Fig. 2 illustrates the detailed workflow of the proposed model.

A. System initialization

In the proposed model, TKA performs the system initialization phase in a secure operational environment to generate the required system parameters for all the involved entities. Initially, with the given security parameter $k > 1$, TKA randomly selects two large prime number $(p, q) > 2^k$, $|p| = |q| = k$ such that $N = (1-p)(1-q)$ relative to which the DCPR is hard in Z_p^* and also selects another random number τ , where $\tau \ll N$. TKA also selects two cryptographic hash functions $H_1, H_2 : \{0, 1\}^* \rightarrow Z_N$ and randomly selects $g \in Z_{N^2}^*$, where the order of g in $Z_{N^2}^*$ is a multiple of N . Afterward, TKA sets (N, g, H_1, H_2) as public-key and $\lambda = lcm(p-1, q-1)$ as private-key of efficient paillier cryptosystem for IoFT layer and ACC. Let $\nu = N+1$ and define the function $L(\nu) = \frac{\nu-1}{N}$.

Further, TKA considers a multiplicative cyclic group $G_1 = \langle g_0 \rangle$ of prime order p with generator g_0 . TKA randomly selects $(a_j, x_i) \in Z_N^*$ and compute $b_j = g_0^{a_j} \mod p$, $y_i = g_0^{x_i} \mod p$ as master-signing key-pairs such that $(ms_{priv_{IT_{ij}}}, ms_{pub_{IT_{ij}}}) = (a_j, b_j)$ and $(ms_{priv_{FS_i}}, ms_{pub_{FS_i}}) = (x_i, y_i)$ for IoFT devices and fog servers respectively. Finally, TKA publishes the system parameter $= (p, q, g, N, g_0, H_1, H_2, b_j, y_i)$ and paillier cryptosystem's public parameter $(\lambda, L(\nu), \kappa)$, where $\kappa = \lambda^{-1}$ to ACC. Moreover, to sake of simplicity, the length of collected crop-status data by each IT_{ij} will not be greater than a defined parameter value $|\zeta| \approx 100$ bit. Then, TKA selects a super linear-sequence $\vec{a} = (a_1, a_2, \dots, a_n)$, where $(a_1, a_2, \dots, a_n) \in p$ and $a_1 = 1$. So, TKA selects a set of sequences (g_1, g_2, \dots, g_n) where $g_i = g^{a_i}, \forall i \in [1..n]$ and shares it as public parameter with all fog servers FS_i .

B. Registration phase

This phase consists of the registration of all IoFT devices and fog servers. Initially, we assume that there are n - numbers of fog servers at Fog layer, which cover a defined area and \mathcal{M} is the number of total IoFT devices at IoFT layer. Further, as described earlier, each i^{th} - fog server covers a group of IoFT devices in a defined area, i.e., IT_{ij} represents as j^{th} - IoFT device in a region of i^{th} - fog server, where $j \in [1..|\vartheta_i|]$, $\vartheta \in \mathcal{M}$, where ϑ is the total IoFT devices in region of a fog server (FS_i). To initiate the registration procedure, all IoFT devices (IT_{ij}) and fog servers (FS_i) share their identities such as $ID_{IT_{ij}}, ID_{FS_i}$, respectively, with TKA. Then, TKA generates the required key-pair for FS_i and IT_{ij} as follows;

- 1) Firstly, TKA randomly selects $s_0, s_1 \in Z_p^*$ and compute $d_0 = g_0^{s_0} \mod p$, $d_1 = g_0^{s_1} \mod p$.
- 2) TKA sets partial-private-key for IT_{ij} and FS_i as $k_0 = s_0 + a_j H_1(ID_{IT_{ij}}, d_0)$ and $k_1 = s_1 + x_i H_1(ID_{FS_i}, d_1)$, respectively.

Then, TKA sends these corresponding partial-private-key to IT_{ij} and FS_i over a secure network channel. Then, IT_{ij} and FS_i verifies these shared partial-private-key with the following validation condition -

$$\begin{aligned} g_0^{k_0} &\stackrel{?}{=} d_0 \cdot b_j^{H_1(ID_{IT_{ij}}, d_0)} \mod p, \\ g_0^{k_1} &\stackrel{?}{=} d_1 \cdot y_i^{H_1(ID_{FS_i}, d_1)} \mod p \end{aligned} \quad (1)$$

If these condition holds, all IoFT devices IT_{ij} accept it; randomly selects $u_0 \in Z_p^*$ and set key-pair as signing-private-key $(MK_{IT_{ij}}) = (k_0, u_0)$ and signing-public-key $(PK_{IT_{ij}}) = (d_0, \mu_0)$, where $\mu_0 = g_0^{u_0} \mod p$ for respective tag-value (σ_{ij}) generation. Similarly, all fog servers FS_i accept the partial-private-key and generate key-pair as signing-private-key $(MK_{FS_i}) = (k_1, u_1)$ and signing-public-key $(PK_{FS_i}) = (d_1, \mu_1)$, where $\mu_1 = g_0^{u_1} \mod p$ by randomly selected $u_1 \in Z_p^*$ for respective tag-value (σ_i) generation.

Further, to initiate the transaction generation and block generation over blockchain at the Fog layer, θ - number of fog servers are randomly selected as authenticated nodes by employing a committee-endorsing mechanism with VRF [38]. Afterward, according to the status of their respective

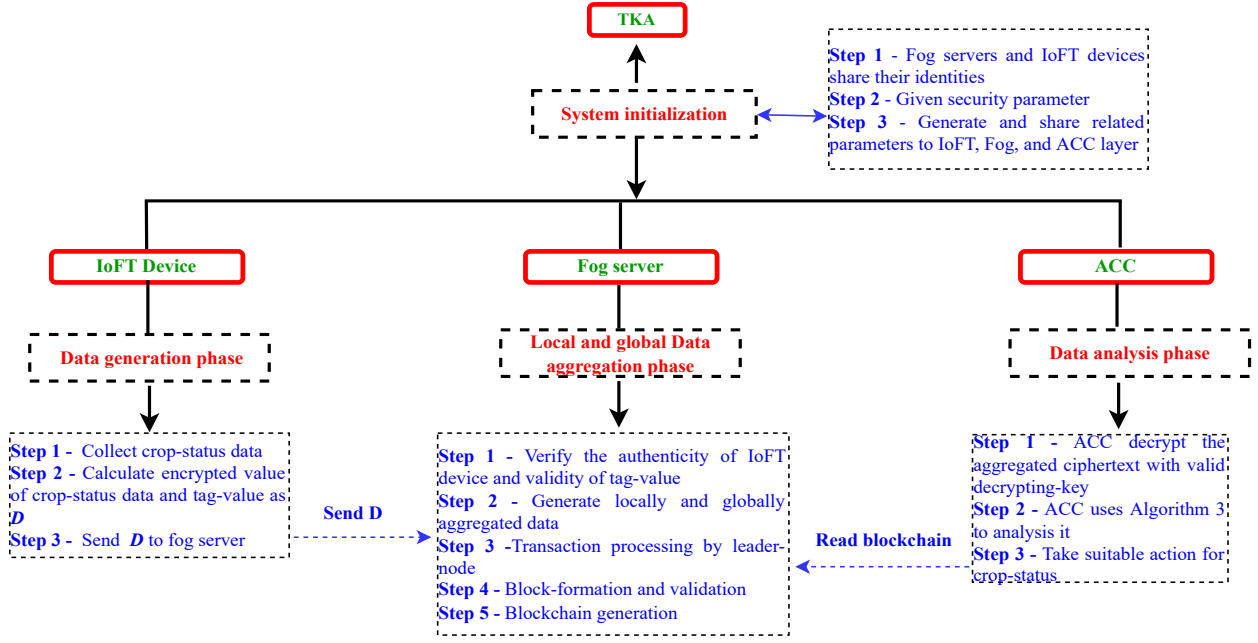


Fig. 2: Illustration of complete data aggregation model

resources, these authorized nodes initiate the order of block right-ownership at various time-slots and broadcast the order to the Fog layer.

C. IoFT layer data generation phase

This phase is executed by all IoFT devices (IT_{ij}) to collect the crop-status data and send it to ACC via Fog layer. Initially, at each time-slot $T_{mt} | mt \in [0, 1, \dots, \varpi]$, (where ϖ is total time-slot in which all the IoFT devices of any i^{th} - region collects crop-status data) each IT_{ij} follows the defined procedure to collect the crop-status data (ms_{ij}) in the following manner;

- 1) IT_{ij} collects the crop-status data of their corresponding agriculture area and generate the encrypted value of collected data (ciphertext) by randomly selecting $r \in_R [0, 1]^\tau$, and set $\gamma_{ij} = [H_2(ms_{ij} || r || T_{mt})]^N$, $M_{s_{ij}} = (ms_{ij} || r)$, then compute the ciphertext as

$$C_{ID_{IT_{ij}}} = g^{M_{s_{ij}}} \cdot \gamma_{ij} \quad \text{mod } N^2$$

- 2) Then, to generate the respective tag-value, IT_{ij} calculates $\chi_{ij} = H_2(C_{ID_{IT_{ij}}} || T_{mt} || PK_{ID_{IT_{ij}}} || ID_{IT_{ij}} || c || c')$, $\delta_{ij} = h - \chi_{ij} \cdot \mu_0 \quad \text{mod } p$, and $w_{ij} = h' - \chi_{ij} \cdot k_0 \quad \text{mod } p$, where $c = g_0^h \quad \text{mod } p$, $c' = g_0^{h'} \quad \text{mod } p$ and $(h, h') \in Z_p^*$. Finally, calculate the tag-value (σ_{ij}) for each collected crop-status data (ms_{ij}) is $\sigma_{ij} = (\chi_{ij}, \delta_{ij}, w_{ij})$.
- 3) Afterward, all the IoFT devices (IT_{ij}) sends the collected crop-status data and tag-value as $D = (C_{ID_{IT_{ij}}}, \sigma_{ij}, T_{mt})$

D. Fog layer data aggregation phase

The fog servers of the Fog layer generally perform the data aggregation through the following sub-phases;

- 1) *Validation of IoFT devices and data*: On receiving $D = (C_{ID_{IT_{ij}}}, \sigma_{ij}, T_{mt})$ from IoFT layer in time-slot T_{mt} , FS_i verifies the authenticity of IoFT devices and validity of tag-value by verifying the validation condition;

$$\begin{aligned} g_0^{k_0} &\stackrel{?}{=} d_0 \cdot b_j^{H_1(ID_{IT_{ij}}, d_0)} \quad \text{mod } p \\ \chi_{ij} &\stackrel{?}{=} H_2(ID_{IT_{ij}}, C_{ID_{IT_{ij}}}, g_0^{\delta_{ij}} \mu^{u_0}, g_0^{w_{ij}}, \\ &\quad (d_0 \cdot b_j^{H_1(ID_{IT_{ij}}, d_0)})^{u_0}, PK_{IT_{ij}}) \end{aligned} \quad (2)$$

- 2) *Fog layer data aggregation phase*: Once these validation condition holds, each FS_i aggregates their local data aggregation for each time-slot T_{mt} with new tag-values for all locally aggregated data as follows;

$$\begin{aligned} C_i &= \prod_{j=1}^{ms_i} C_{ID_{IT_{ij}}} \quad \text{mod } N^2, \\ &= \prod_{j=1}^{ms_i} \left(g^{M_{s_{ij}}} \cdot [H_2(ms_{ij} || r || T_{mt})]^N \right) \quad \text{mod } N^2 \end{aligned} \quad (3)$$

Also, FS_i computes the new tag-values by randomly selecting $(e, e') \in Z_p^*$ and calculates $m = g_0^e \quad \text{mod } p$, $m' = g_0^{e'} \quad \text{mod } p$. Then, sets $v_i = H_2(C_{ID_{FS_i}} || T_{mt} || PK_{ID_{FS_i}} || ID_{FS_i} || e || e')$, $\varphi_i = m - v_i \cdot \mu_1 \quad \text{mod } p$, and $\psi_i = m' - v_i \cdot k_1 \quad \text{mod } p$. So, the corresponding new tag-values for all locally aggregated data (C_i) is $\sigma_i = (v_i, \varphi_i, \psi_i)$.

Afterward, every FS_i pack these results into as validation-value ($V_{S_i} = (C_i || \sigma_i || ID_{FS_i} || T_{mt})$) and sends it to authorized nodes FS_θ , where $\theta \in [1, n]$. Note that, in our proposed model θ - represents as the number of fog servers, which are randomly selected by employing committee-endorsing mechanism with VRF as authenticated nodes. Also, one of the authorized node randomly selects as a leader-node with block

generation right over blockchain for time-slot (T_{mt}). Before initiating the block generation on the Fog layer blockchain network, the leader node require to verify the authenticity of fog servers and the validity of corresponding new tag-value (σ_i) by verifying the validation condition;

$$\begin{aligned} g_1^{k_1} &\stackrel{?}{=} d_1 \cdot y_i^{H_1(ID_{FS_i}, d_1)} \pmod p \\ v_i &\stackrel{?}{=} H_2(ID_{FS_i}, C_i, g_0^{\varphi_i}, g_0^{\psi_i} \mu^{u_1}, \\ &\quad (d_1 \cdot y_i^{H_1(ID_{FS_i}, d_1)})^{u_1}, PK_{FS_i}) \end{aligned} \quad (4)$$

If these condition holds, the leader-node executes the global data aggregation operation to generate aggregate encrypted-data (C_{AS}) and aggregated tag-value (σ_{FS}) as follows;

$$\begin{aligned} \sigma_{AS} &= \prod_{i=1}^n \sigma_i = \prod_{i=1}^n (v_i, \varphi_i, \psi_i), \\ C_{AS} &= \prod_{i=1}^n C_i \pmod{N^2} = \prod_{i=1}^n (g^{Ms_{ij}} \cdot \gamma_{ij} \pmod{N^2}) \\ &= \left(g^{\sum_{j=1}^{\theta_1} ms_{1j}} \cdot g^{\sum_{j=1}^{\theta_2} ms_{2j}} \cdot \dots \cdot g^{\sum_{j=1}^{\theta_n} ms_{nj}} \right) \\ &\quad \times \prod_{i=1}^n (\gamma_{ij} \pmod{N^2}) \end{aligned}$$

Since, (g_1, g_2, \dots, g_n) satisfies the defined condition $g_i = g^{a_i}, \forall i \in [1 \dots n]$, the above derived condition can be formulated as;

$$\begin{aligned} &= \left(g^{\sum_{j=1}^{\theta_1} ms_{1j}} \cdot g^{\sum_{j=1}^{\theta_2} ms_{2j}} \cdot \dots \cdot g^{\sum_{j=1}^{\theta_n} ms_{nj}} \right) \\ &\quad \times \prod_{i=1}^n (\gamma_{ij} \pmod{N^2}) \\ &= g^{a_1 \sum_{j=1}^{\theta_1} ms_{1j} + a_2 \sum_{j=1}^{\theta_2} ms_{2j} + \dots + a_n \sum_{j=1}^{\theta_n} ms_{nj}} \\ &\quad \times \prod_{i=1}^n [H_2(ms_{ij} || r || T_{mt})]^N \pmod{N^2} \end{aligned}$$

Afterward, leader-node sends the aggregated encrypted-data (C_{AS}) with σ_i to ACC.

3) **Block generation (B_j):** After successfully generating the aggregate encrypted-data (C_{AS}), leader-node generates the respective transaction (T_{x_i}) and block (B_j) for C_{AS} with validation-value (V_{S_i}). Initially, the transaction generation procedure follows basic protocol with time-slot (T_{mt}) and leader-node generate the hash-tuple $B_j = \langle Header_j, D_j \rangle$, where $Header_j = \langle T_{mt_j}, st_j, iB, H_2(D_j), \pi_j \rangle$ and D_j represent the Block-data, $T_{mt_j} \in (T_{mt_1}, T_{mt_2}, \dots, T_{mt_n})$ represents n different time-slots, st_j represents Previous-Block's Hash [$H_2(Header_{j-1})$], iB is the initial state of the Block-data, and π_j represents Nonce-value for consensus algorithm (PoW). Moreover, the tuple $\langle FS_{\xi} | \forall \xi \in \theta, T_{mt} \rangle$, represent that FS_{ξ} is a leader-node at time-slot (T_{mt}) to validate the generated block B_j .

So, initially, leader-node (FS_{ξ}) generates the respective block B_j with transaction (T_{x_i}) = $H_2(V_{S_i} || Nonce - value || ts)$, where ts is timestamp value of transaction (T_{x_i}). Then, the leader-node (FS_{ξ}) broadcast the block B_j to all

authorized nodes (FS_{θ}) over the Fog layer for the proper validation. After this, the ValidateBlock algorithm (*Algorithm 1*) verifies the validity of Block-data (B_j) according to defined rules. Then, it verifies the leader's validity and finally verifies the Nonce_value (π_j) for the defined Difficulty (\mathcal{D}) to complete PoW consensus. Now, if $ValidateBlock(B_j)$ algorithm returns 1; update the chain with new block (B_j) according to specified blockchain protocol [14].

Algorithm 1: Block-Validation Algorithm;
ValidateBlock(B_j)

- 1 Parse $B_j = \langle Header_j, D_j \rangle$,
where $Header_j = \langle T_{mt_j}, st_j, iB, H_2(D_j), \pi_j \rangle$;
 - 2 Validate D_j , **if** invalid return 0;
 - 3 Validate FS_{ξ} , **if** invalid return 0, where FS_{ξ} is leader for current T_{mt} ;
 - 4 Validate π_j , **if** invalid return 0;
 - 5 **else, return 1.**
-

4) **Redactability of collected data:** In some cases, IoFT devices become unstable. They may collect and send erroneous data to corresponding fog servers, and Fog layer sends it to ACC with an immutable transaction of a block over the blockchain network. Further, in blockchain-assisted models, after a severe 51% attack, malicious fog servers could lead to modification attacks and collusion attacks with possibly fatal results regarding crop-status data at ACC. So, in the above-mentioned cases, ACC will not be able to take the correct required action against any issues with crops. Thus, we employ redactable blockchain architecture which enables the authorized nodes (FS_{θ}) and leader-node (FS_{ξ}) to redact the corresponding block having aggregated encrypted-data (C_{AS}) as follows. Firstly, we define two application specific functions, $Cpt(Chain, T_{mt}, FS_{\xi})$ and $Verify(Chain, T_{mt}, \rho, Prf)$. First function $Cpt(Chain, T_{mt}, FS_{\xi})$ to verify the validity of newly selected leader-nodes in voting-period of time-slot T_{mt} and produce output (Prf, ρ), where Prf is proof of FS_{ξ} for T_{mt} and ρ is voting-weight of leader-node FS_{ξ} in FS_{θ} . Second function is $Verify(Chain, T_{mt}, \rho, Prf)$ to verify the (Prf, ρ).

- 1) **Initialize redaction** - FS_{ξ} initializes redaction to a block B_j in blockchain. Then, FS_{ξ} parses $\langle Header_j, D_j \rangle$ and replace D_j with D_j^* , where D_j^* is Block-data, $Header_j$ with $Header_j^*$ such as $Header_j^* = \langle T_{mt_j}, st_j, iB_j, \pi_j, H_2(D_j^*) \rangle$ to create new block $B_j^* = \langle Header_j^*, D_j^* \rangle$ and broadcast to blockchain. Afterward, elects a leader FS_{ξ} by leader-selection algorithm to validate B_j^* whether it is valid candidate redacting block or not and stores it in their pool in the following manner. (i) FS_{ξ} verifies the expiration-time (t_p) since every candidate redacting block B_j^* has a defined t_p , (ii) if B_j^* is valid, FS_{ξ} verifies $RP(Chain, B_j^*, sl_j) \stackrel{?}{=} 1$ and remove B_j^* from pool, where $RP(Chain, B_j^*, sl_j)$ represents the redaction-policy to verify the Nonce-value (π_j) for the defined Difficulty (\mathcal{D}). Initially, $ValidateCedt()$ checks the validity of B_j^* , then, verifies the link between B_{j-1} and B_{j+1} . Finally, B_j^* is a valid candidate editing block

if $ValidateCedt = 1$. *Algorithm 2* illustrates the validation of candidate editing block B_j^* .

- 2) **Validation procedure of B_j^*** - For candidate block B_j^* , FS_ξ submits solution of Difficulty \mathcal{D} and FS_θ verifies their validity in T'_{mt_j} with $Cpt(Chain, T'_{mt_j}, FS_\theta)$ according to defined PoW, where T'_{mt_j} represents the current time-slot. FS_ξ outputs $[(\rho, Prf), \rho \neq 0]$ and broadcast (ρ, Prf) with Nonce-value on $H_2(B_j^*)$ as their respective solutions. Afterward, FS_ξ verifies through *Algorithm 1*, initially, verifies the number of votes on $H_2(B_j^*)$ by $RP(Chain, B_j^*, T'_{mt_j})$ is enough or not. Particularly, firstly, it verifies the Nonce-value (π_j), confirms the voting-right, and voting-number ρ of the voters defined by $Verify(Chain, T_{mt}, \rho, Prf)$. Finally, if $RP(Chain, B_j^*, T'_{mt_j}) = 1$, FS_θ updates the blockchain $Chain$ with B_j^* . *Algorithm 3* illustrates the validation procedure of B_j^* .

Algorithm 2: Candidate Block-Validation Algorithm;
ValidateCedt(Chain, B_j^)*

- 1 Parse $B_j^* = \langle Header_j^*, D_j^* \rangle$, where
 $Header_j^* = \langle T_{mt_j}, st_j, H_2(D_j^*, iB_j, \pi_j) \rangle$;
 - 2 **if** $ValidateBlock(B_j^*) = 0$, **then return 0**;
 - 3 Parse $B_{j-1} = \langle Header_{j-1}, D_{j-1} \rangle$, where
 $Header_{j-1} = (T_{mt_{j-1}}, st_{j-1}, \pi_{j-1}, iB_{j-1}, H_2(D_{j-1}))$
 - 4 Parse $B_{j+1} = \langle Header_{j+1}, D_{j+1} \rangle$, where
 $Header_{j+1} = (T_{mt_{j+1}}, st_{j+1}, \pi_{j+1}, iB_{j+1}, H_2(D_{j+1}))$
 - 5 **if** $st_j = H_2(T_{mt_{j-1}}, st_{j-1}, iB_{j-1}, iB_{j-1}, \pi_{j-1})$ and
 $st_{j+1} = H_2(T_{mt_j}, st_j, iB_j, iB_j, \pi_j)$, **then return 1**;
 - 6 **else, return 0**.
-

Algorithm 3: Validation - procedure of B_j^*

- 1 Set $\rho = 0$, $Prf = 0$, Time-slot = T_{mt} ;
 - 2 **while** Time-slot $\leq T_{mt} + s - 1$, where s is required slots for votes distribution for candidate block;
 - 3 Parse $Chain = (B_1, \dots, B_\phi)$, assume ϕ be the latest confirm height on blockchain;
 - 4 Parse $(B_j)_{Time} = (Time - slot, st_j, H_2(D_j, iB, \pi_j))$;
 - 5 **if** FS_θ find correct Nonce-value (π_j),
 $H_2(Time - slot, st_j, H_2(D_j), iB, \pi_j) < \mathcal{D}$;
 - 6 **then** set $\rho = \rho + 1$, $Prf = Prf \cup (Time - slot, st_j, H_2(D_j), iB, \pi_j)$;
 - 7 **end, while**;
 - 8 **return** (ρ, Prf) .
-

E. Data analysis at ACC

ACC reads the blockchain records every η minutes, where $\eta \subset t_{mt}$. Then, ACC decrypt the global aggregated data (C_{AS}) through defined decryption method of efficient paillier

cryptosystem. Meanwhile, we describe symbols \mathcal{S} and \mathcal{P} to facilitate the decryption of aggregated ciphertext;

$$\mathcal{S} = a_1 \sum_{j=1}^{\vartheta_1} ms_{1j} + a_2 \sum_{j=1}^{\vartheta_2} ms_{2j} + \dots + a_n \sum_{j=1}^{\vartheta_n} ms_{nj}, \quad (5)$$

$$\mathcal{P} = \prod_{i=1}^n \gamma_{ij}$$

The aggregated ciphertext (C_{AS}) can be converted into the form of $C_{AS} = g^{\mathcal{S}} \cdot \mathcal{P} \pmod{N^2}$.

However, the converted aggregated ciphertext still adheres to the paillier encryption format. So, ACC uses the paillier decryption key and $L(\nu)$ to perform the decryption to get the aggregated data

$$\mathcal{M}' = \frac{L(C_{AS}^{\lambda} \pmod{N^2})}{L(g^{\lambda} \pmod{N^2})} \quad (6)$$

and parses the Ms_{ij} as $(ms_{ij}||r)$. Then, ACC computes $\gamma_{ij} = g^{-Ms_{ij}} \cdot C_{ID_{IT_{ij}}} \pmod{N^2}$ and verify if $\gamma_{ij} \stackrel{?}{=} H_2[(ms_{ij}||r)^N \pmod{N^2}]$, return ms_{ij} , else return invalid.

Finally, Horner's rule [40] is used to complete the analysis of aggregated collected data and get fine-grained aggregation results in *Algorithm 4*. The algorithm generates the coefficient PA_j , which describes the crop-status data of one IoFT device;

$$PA_j = \sum_{j=1}^{\vartheta_i} ms_{ij}, \forall i \in [1, n] \quad (7)$$

Once ACC obtains the crop-status data of each IoFT device of the defined region of corresponding fog servers. Afterward, these fine-grained data can be explored to analyze the real-time status of crops and also take suitable action whenever required.

Algorithm 4: Aggregated Area-Report Extraction

Input: \mathcal{S} and \mathcal{P}

Output: crop-status $PA_j, j \in [1, \vartheta]$ of each IoFT device

- 1 **Begin**
 - 2 $q_0 \leftarrow \mathcal{S}/\mathcal{P}$, $a_1 = \mathcal{P}^{1j}$, $a_2 = \mathcal{P}^{2j}$, ..., $a_n = \mathcal{P}^{nj}$,
 $q_0 \leftarrow PA_1 + \mathcal{P}^1 PA_2, \dots, \mathcal{P}^{n-1} PA_n$;
 - 3 **for** $j \leftarrow 1$ to ϑ_i **do**;
 - 4 $PA_j \leftarrow q_{j-1} \pmod{\mathcal{P}}$;
 - 5 $q_j \leftarrow q_{j-1} \pmod{\mathcal{P}}$;
 - 6 **End for**
 - 7 Obtain $(PA_1, PA_2, \dots, PA_\vartheta)$
 - 8 **End**
-

VI. SECURITY ANALYSIS

In this section, a detailed explanation about the security analysis of the proposed data aggregation model is demonstrated with concrete proofs in terms of various security theorems.

Theorem 1. (Confidentiality and Privacy-preserving) *The proposed model guarantees the complete confidentiality and*

1 *privacy-preserving against malicious Fog servers (FS_i) and*
 2 *ACC.*

3
 4 *Proof.* The proposed model ensures computationally in-
 5 tractability for confidentiality against any malicious fog servers
 6 (FS_i) at Fog layer and ACC with proper privacy-preserving
 7 as follows;

8
 9 **Case 1 - Confidentiality against FS_i :-** We employ the
 10 efficient IND-CCA2 secure paillier cryptosystem to encrypt
 11 the crop-status data C_i in order to ensure security against data
 12 confidentiality during the data aggregation phase. Since, after
 13 passing the validation condition (Eq. 2), the encrypted data
 14 $C_{IDIT_{ij}}$ can be considered as valid. Moreover, the compu-
 15 tationally intractability of DCPR in a random oracle against
 16 IND-CCA2 also proven the complete security of efficient
 17 paillier cryptosystem [13]. So, it is computationally hard for
 18 malicious fog server (FS_i) to decrypt the shared encrypted
 19 data $C_{IDIT_{ij}}$ and obtain the real data (ms_{ij}). Simultaneously,
 20 due to computationally intractability of encryption method,
 21 any authorized nodes (FS_θ) is unable to decrypt the aggre-
 22 gated ciphertext (C_{AS}). Thus, the proposed model ensures the
 23 data confidentiality of crop-status data against malicious FS_i .

24
 25 **Case 2 - Confidentiality against ACC :-** To show the
 26 computationally infeasibility of data confidentiality at ACC,
 27 assume a PPT adversary (A_I) colludes with ACC and obtains
 28 the required decryption key (λ, κ). However, if A_I tries to
 29 decrypt C_i , there is only option to obtain the randomly
 30 selected value (r) directly, since, the collected crop-status
 31 data (ms_{ij}) is blinded by r , i.e., $Ms_{ij} = (ms_{ij}||r)$. So,
 32 A_I cannot be able to obtain the value of ($ms_{ij}||r$) with
 33 the given tuple $\langle g, g^{Ms_{ij}} \rangle$, until the CDH-problem is com-
 34 putationally hard in G_1 [39]. On the other hand, it is ob-
 35 viously computationally hard to break the security of Hash-
 36 function in $\gamma_{ij} = [H_2(ms_{ij}||r||T_{mt})]^N$ to form real value
 37 of ms_{ij} [41]. Moreover, although A_I can be able to get the
 38 aggregated crop-status data $g^{a_1 \sum_{j=1}^{\theta_1} ms_{1j} + a_2 \sum_{j=1}^{\theta_2} ms_{2j} + \dots + a_n}$
 39 through successful intruding the ACC, still A_I can not be able
 40 to obtain the every IoFT devices' collected data and link to
 41 its respective region of FS_i , due to solving a NPC-problem
 42 [42]. Moreover, even if A_I manages to gain access to the fog
 43 server, A_I will be unable to access any private information.

44 So, none of the PPT adversary A_I can collude with ACC to
 45 attack the confidentiality of aggregated data and also privacy
 46 of IoFT devices. Therefore, the proposed data aggregation
 47 model guarantees the proper data confidentiality and privacy-
 48 preserving of IoFT devices. ■

49 **Theorem 2. (Authentication and Integrity)** *The proposed*
 50 *model guarantees the proper authentication for transfer be-*
 51 *tween IoFT layer, Fog layer and data integrity for collected*
 52 *data.*

53
 54 *Proof. 1. Source authentication -* To show the proper
 55 authentication of the model, we consider the communi-
 56 cation between IoFT layer-to-Fog layer and Fog layer-to-
 57 ACC. Initially, we consider the communication between IoFT
 58 layer-to-Fog layer. Upon receiving $D = (C_{IDIT_{ij}}, \sigma_{ij}, T_{mt})$
 59 from IoFT devices in time-slot (T_{mt}), FS_i verifies the

authenticity of IoFT devices with verification condition -
 $g_0^{k_0} \stackrel{?}{=} d_0 \cdot b_j^{H_1(IDIT_{ij}, d_0)} \pmod p$; and then check the va-
 lidity of tag-value (σ_{ij}) with corresponding public-key as
 follows - $\chi_{ij} \stackrel{?}{=} H_2(IDIT_{ij}, C_{IDIT_{ij}}, g_0^{\delta_{ij}} \mu^{u_0}, g_0^{w_{ij}}, (d_0 \cdot$
 $b_j^{H_1(IDIT_{ij}, d_0)})^{u_0}, PK_{IT_{ij}})$.

However, a PPT adversary (A_{II}) can attack as Type-I
 adversary, i.e., any Type-I adversary can replace the defined
 public-key, i.e. IoFT devices' public-key ($PK_{IT_{ij}} = (d_0, \mu_0)$),
 but does not have any information about signing-private-key
 ($MK_{IT_{ij}} = (k_0, u_0)$). Then, A_{II} tries to pass the validation
 procedure (Eq. 1) with replaced public-key by checking the
 validity of tag-value (σ_{ij}). Now, we need to provide the
 proof for which only the real key-pair is required to pass the
 validation condition (Eq. 1).

To prove the security of authentication procedure and tag-
 value, we design an algorithm \mathcal{B} , i.e., \mathcal{B} uses A_{II} as black
 box to find the solution of DL-problem in polynomial time
 with limited power. Initially, \mathcal{B} is given the prime num-
 ber p, q and a DL-problem instance ($g_0, R = g_0^A$). The
 aim of algorithm \mathcal{B} is to find A , i.e., $R = g_0^A$. \mathcal{B} also
 initialize A_{II} with the signing-key-pairs and given system
 parameter = $(p, q, g, N, g_0, H_1, H_2, b_j, y_i)$ to A_{II} . So, \mathcal{B} re-
 sets A_{II} to use an oracle-query $H_2(\cdot)$ on the tag-string =
 $H_2(C_{IDIT_{ij}} || T_{mt} || PK_{IDIT_{ij}} || IDIT_{ij} || c || c')$, $\delta_{ij} = h - \chi_{ij} \cdot \mu_0$
 mod p for getting χ_{ij} , according to the tag-generation (σ_{ij})
 procedure. Then, \mathcal{B} rearrange the random-oracle for getting
 $\chi_{ij}^* \in Z^*$, i.e., $\chi_{ij}^* \neq \chi_{ij}$ and proceeds to simulate A_{II} .
 Suppose, if A_{II} is an efficient forger in the aforementioned
 interactions, \mathcal{B} can get two valid tag-values such as $\sigma_{ij} =$
 $(\chi_{ij}, \delta_{ij}, w_{ij})$ and $\sigma_{ij}^* = (\chi_{ij}^*, \delta_{ij}^*, w_{ij}^*)$ with the defined
 condition that $\chi_{ij}^* \neq \chi_{ij}$ which satisfies;

$$g_0^{w_{ij}} \left(d_0 \cdot b_j^{H_1(IDIT_{ij}, d_0)} \right)^{\chi_{ij}} = g_0^{w_{ij}^*} \left(d_0 \cdot b_j^{H_1(IDIT_{ij}, d_0)} \right)^{\chi_{ij}^*}$$

Afterward, \mathcal{B} can compute;

$$\log_{g_0}(d_0) = \left[\frac{(w_{ij} - w_{ij}^*)}{(\chi_{ij} - \chi_{ij}^*)} \right] - s_0 (H_1(IDIT_{ij}, d_0)) \pmod{N^2}$$

Where, N^2 is a public prime integer, i.e., $N = (1 - p)(1 - q)$
 and $(w_{ij} - w_{ij}^*) / (\chi_{ij} - \chi_{ij}^*)$ is also prime to N^2 . So, \mathcal{B}
 can find solution of DL-problem to get-value of d_0 . However,
 clearly, it is contradicting with the intractability to solve the
 hard DL-problem in polynomial time. So, the proposed model
 guarantees the proper authenticity with security against false
 data injection attack by A_{II} between communication IoFT
 layer-to-Fog layer.

2. **Integrity -** As defined earlier, on receiving $D =$
 $(C_{IDIT_{ij}}, \sigma_{ij}, T_{mt})$ from IoFT devices in time-slot (T_{mt}), FS_i
 verifies the validation of tag-value (σ_{ij}) along with authenticity
 of IoFT devices. Also, it is discussed above only the regis-
 tered IoFT devices have knowledge of real signing-public-key
 $(PK_{IT_{ij}} = (d_0, \mu_0))$ linked to respective signing-private-key
 $MK_{IT_{ij}} = (k_0, u_0)$, it can generate a valid tag-value which
 passes the validation condition (Eq. 2). As A_{II} can not solve
 the hard DL-problem corresponding to pass the validation
 condition with modified tag-value (σ_{ij}^*) in polynomial time

with limited power, so, A_{II} can not pass the verification condition with modified encrypted crop-status data ($C_{ID_{IT_{ij}}}$). Furthermore, each of the collected crop-status data are marked with time-stamp (T_{mt}) value, which is also included into $D = (C_{ID_{IT_{ij}}}, \sigma_{ij}, T_{mt})$. So, the current time-stamp value guarantees that A_{II} can not perform any reply attack also on the proposed model.

Afterward, the similar verification procedure also performed by leader-node to check the authenticity and validity of aggregated data (C_i) and new tag-value (σ_i) with time-stamp (T_{mt}). Therefore, the proposed model also ensures the proper integrity of the collected crop-status data. ■

Theorem 3. (Data mining attack) *The proposed model guarantees the security against the malicious data mining attack.*

Proof. In the proposed model, we assure that a PPT adversary (A_{II}) tries to launch a data mining attack against target IoFT device at IoFT layer. Here, under the following conditions, we will show the resistance of the proposed data aggregation model against malicious data mining attack;

Case 1: We consider the initial condition, where A_{II} blocks the communication from one target IoFT device, i.e., $IoFT_1$ to respective fog server (FS_i). For such condition, $IoFT_1$ cannot share the respective tuple, i.e., $k_0 = s_0 + a_j H_1(ID_{IT_{ij}}, d_0)$ and $d_0 = g_{s_0}^0 \bmod p$ to FS_i . So, IoFT layer sends the collected crop-status data of ($j - 1$) IoFT device as $D^* = (C_{ID_{IT_{ij}}}, \sigma_{ij}^*, T_{mt})$. Then, FS_i can only able to aggregate ($j - 1$) IoFT device's data at Fog layer as;

$$C_i^* = \prod_{j=1}^{ms_i} C_{ID_{IT_{ij}}}^* \bmod N^2$$

after holding the validation condition (Eq. 2) with tuple (k_0^*, d_0^*) such as;

$$\prod_{j=2}^{\vartheta_i} g_0^{k_0^*} \stackrel{?}{=} d_0^* \cdot b_j^{H_1(ID_{IT_{ij}}, d_0)} \bmod p$$

However, A_{II} cannot be able to pass the validation condition (Eq. 2) with D^* and (k_0^*, d_0^*) . Since, it is not possible to hold validation condition (Eq. 2), $g_0^{k_0^*} \stackrel{?}{=} d_0^* \cdot b_j^{H_1(ID_{IT_{ij}}, d_0)} \bmod p$, where $b_1 \neq 0 \bmod p$. So, A_{II} cannot be attempt any malicious data mining attack to block communication between IoFT layer-to-Fog layer by targeting any IoFT device.

Case 2: Now, we consider an internal attack from A_{II} , then any of the IoFT device collude with A_{II} and block the communication between IoFT layer to Fog layer. For such condition, A_{II} interrupt the previously collected data by one IoFT device ($IoFT_1$), parse the value of $g_0^{k_0}$ with b_2 , and modifies the ciphertext of their collected data ($C_{ID_{IT_{ij}}}$) and σ_{ij} . On receiving the encrypted crop-status data of $IoFT_2$ from IoFT layer, respective fog server (FS_i) checks the validity it through Eq 4. However, due to computational intractability of DL-problem in G_1 , it is not possible to pass the validation condition with modified values of $(C_{ID_{IT_{ij}}}, \sigma_{ij})$. So, in such

condition, A_{II} cannot launch any malicious data mining attack as internal attack. ■

Theorem 4. (Collusion and Modification attack) *The proposed model also guarantees the security of collected and transferred crop-status data on blockchain against any modification and collusion attack.*

Proof. As described earlier, collected data of each IoFT devices' is integrated into a block over blockchain at Fog layer. So, whenever any malicious ACC wants to modify to existing collected data with modified values for some personal benefits, then, only option is to break the security of redactable blockchain architecture. Specifically, the malicious ACC incentivized the group of authorized nodes ($FS_d | d \in [1, \dots, \theta]$) to modify any block B_j with malicious block B_{j^*} . To perform such attack, we assume that the leader-node can find a solution of the defined Difficulty (\mathcal{D}) to complete the PoW consensus for validation of malicious block B_{j^*} in at most ϑ time-slots earlier than honest fog servers. Initially, another assumption that, $\beta = \frac{1}{2} + \varepsilon$ fraction of fog servers ($FS_l | \forall l \in [1, \dots, \theta]$) are honest, where $\varepsilon \in [0, \frac{1}{2}]$.

Let $\iota = \frac{\mathcal{D}}{2^t}$ and $\iota = \frac{\mathcal{D}}{2^t}(1 - \beta)t$ represent the expected solutions of the defined Difficulty (\mathcal{D}) found by the honest fog servers and incentivized malicious fog servers FS_d in every time-slot respectively, where l is the length of Hash-function $H_2(\cdot)$. We also represent maximum number of solutions for the defined Difficulty (\mathcal{D}) found by incentivized malicious fog servers FS_d in time-slot $\in [T_{mt} - \vartheta, T_{mt} + s - 1]$ as N_e and minimum number of solutions for the defined Difficulty (\mathcal{D}) found by incentivized malicious fog servers FS_d in time-slot $\in [T_{mt}, T_{mt} + s - 1]$ by N_f , respectively. Also, according to *chernoff bound* [43], it holds the conditions such as $N_e \leq [(1 + \varrho)\iota(s + \vartheta)] | \forall \varrho > 0$, except with negligible probability as;

$$P_1 = e^{-\frac{(\varrho \cdot \min(\varrho, 1) \cdot \iota \cdot (s + \vartheta))}{3}}$$

and $N_f \leq [(1 - \varrho)\iota s | \forall \varrho \in [0, 1]]$, except with negligible probability $P_2 = e^{-\frac{\varrho^2 s \iota}{2}}$.

Then, initially, if we set the leader node (FS_ξ) are honest in s - time-slot to break the security of blockchain, the malicious ACC assures the condition $N_e < N_f$. Also, the malicious ACC should hold the condition $(1 + \varrho)\iota(s + \vartheta) < (1 - \varrho)\iota s$ and obtain the condition;

$$w > \frac{\vartheta}{[(1 - \varrho)\beta / ((1 - \varrho)(1 - \beta))] - 1}$$

Moreover, according to Proof-of-consistency property [14], none of the adversary can 'withhold' a block for "too long" and make it to the chain. The malicious ACC should set s be the longest number of time-slots that is incentivized FS_d to withhold the target candidate editing block B_{j^*} . Then, malicious ACC considers the case whenever FS_d withholds some blocks, B_0 new blocks are mined in the longest chain with B_{j^*} where B_0 represent common prefix parameter [14]. According to common prefix property, these withholding blocks can never present in longest chain of honest fog server. So, s should be less than the defined minimum time, FS_d takes for longest valid chain to grow by atleast B_0 blocks. Also, according

to the chain-growth property [14], malicious ACC and FS_d should hold the condition $s \approx \frac{B_0}{i'}$, where $i' = \frac{D'}{2t}\beta t$, and D' is the *Difficulty* for the defined PoW consensus that atleast one incentivized user can find the solution for D' at every time-slot, i.e., $\frac{D'}{2t}t = 1$.

However, the above mentioned conditions are contradictory to common prefix property and chain-growth property with limited power of a group of incentivized fog server which is less than 51% of the overall blockchain network. So, malicious ACC with incentivized fog servers cannot perform any modification and collusion attack on collected data to gain any advantages. ■

VII. PERFORMANCE ANALYSIS

In this section, we demonstrate the proposed model's performance analysis regarding - security functionalities, theoretical, and experimental analysis. Moreover, to provide a detailed analysis in terms of efficiency, we also provide a significant comparison of the proposed model with existing models [25], [31], [34], [35].

A. Functional comparison

Table I demonstrates the functionality comparison of the proposed model with existing models [25], [31], [34], [35] in terms of functionalities and features. All the compared existing data aggregation models [25], [31], [34], [35] are computationally intractable in terms of data integrity and privacy-preserving. So, the proposed data aggregation model and other compared existing data aggregation models can guarantee the proper correctness for all collected crop-status data and privacy-preserving for all IoFT devices. To attain high-level immutability for all collected crop-status data, the existing models [31], [34], and the proposed data aggregation model utilizes the security of blockchain technology. However, in considering the redaction or editing of a block's data over blockchain, only the proposed model employs the redactable blockchain framework to provide flexibility against any erroneous data collection by IoFT devices. Further, while transferring the collected data, the proper source authentication shows the computational infeasibility of communication links and all involved entities against - false data injection attacks in the proposed model and existing model [34]. Considering data confidentiality, most of the existing data aggregation models provide security against it. However, only the work introduced by [34], and the proposed model consider IND-CCA2 security against chosen ciphertext attack and also infeasibility against malicious data mining attack by internal or external adversaries. Also, the proposed model considers the collusion attack to follow the standard security model. Thus, none of the malicious entities can collude to perform collusion attack to break the security of the model.

Moreover, only the proposed model and the work compiled by [34] provide the fault-tolerant to support the robust nature of the data aggregation models. So, as observed from Table I, the proposed data aggregation model can strongly support the standard functionalities with the proper security standard.

B. Theoretical analysis

Primarily, we consider the cryptographic operations involved in the registration, data aggregation, and data analysis phases to analyze the proposed model's efficiency. So, we summarize the theoretical analysis in terms of computational and communication overhead, which is demonstrated in the following manner.

1) *Communication overhead*: Mainly, in the proposed model, we consider the communication overhead between IoFT layer-to-Fog layer and Fog layer-to-ACC. For the sake of simplicity, we take into account that the only one region to analyze the communication overhead of the whole data aggregation procedure. Initially, the IoFT devices of respective agriculture region generates the crop-status data $D = (C_{ID_{IT_{ij}}}, \sigma_{ij}, T_{mt})$ and sends it to corresponding fog server (FS_i). The size of the collected data is $-mw_i(|C_{ID_{IT_{ij}}}| + |\sigma_{ij}|)$, where mw_i is number of aggregated encrypted-data ciphertext at one instance of time-slot. So, the overall communication overhead between IoFT layer-to-Fog layer is $(|N| + |T_{mt}| + 2|p| + |G_1|)$. Afterward, we analyze the communication overhead between Fog servers and leader-node at Fog layer, FS_i performs the local data aggregation to aggregate all the ciphertext of collected data, generates the corresponding transaction (Tx_i) with validation-value (Vs_i) and broadcast it to authorized nodes (FS_θ) to validate it. Then, the Fog layer will generate overhead approximately $(|N| + |p| + |G_1| + |T_{mt}|)$. Similarly, for the works [34], the IoT devices require extra overhead $- (3|N| + 3|p| + |G_1| + 2|T_{mt}|)$ to send collected encrypted data to Fog layer, due to hash-chain based authentication method. Also, the communication overhead between Fog layer-to-cloud center is also high $(2|N| + 3|p| + |G_1| + 2|T_{mt}|)$. Similarly, the works [31], communication overhead from IoT layer-to-Fog layer and Fog layer-to-Cloud center is $(4|N| + 2|p| + |G_1| + 2|T_{mt}|)$ and $(2|N| + 2|p| + |G_1| + 2|T_{mt}|)$, respectively.

Further, the works introduced by [25], [35] also requires some high communication overhead to send collected data from IoT device - to - fog devices is $(2|N| + 2|p| + |G_1|)$ and $(2|N| + |p| + |G_1|)$, respectively. Similarly, from Fog layer - to - Cloud center is $(3|N| + 3|p| + |G_1| + |T_{mt}|)$ and $(2|N| + |p| + |G_1| + |T_{mt}|)$, respectively. So, the efficient communication overhead of the proposed model provides over other existing models provides more feasibility.

2) *Computational overhead*: The computational overhead of the proposed data aggregation model is mainly expressed in terms of cryptographic operations, i.e., exponential operation in $G_1 - (E_N)$, pairing operation $- (P)$, hash-function $- (E_{H_i})$, modular multiplication in $G_1 - (E_m)$, and exponential operation in $Z_{N^2}^* - (E_{N^2})$. Also, to calculate the computational overhead of the proposed model and other existing models [25], [31], [34], [35], under the defined standard security level of 80 - bit, we select a Type-III pairing, i.e., $e : G_1 \times G_2 \rightarrow G_2$ on a super-singular Elliptic curve $E : y^2 = x^3 + ax + b \pmod p$ on field F_p , where b and p both are random prime number on field $F_p \approx 160$ bit. Moreover, we employ PBC - (0.5.14) library to implement all the above mentioned cryptographic operations. It is important to note that the modular exponential

TABLE II: Comparative analysis in the functionalities of different schemes and proposed model

Schemes	Functional characteristics							
	Source authentication	Redactability in Blockchain	IND-CCA2 secure Data confidentiality	Resistance against data mining attack	Resistance against Collusion attack	Privacy-preserving	Data - integrity	Robustness
Lu et al. [34]	Y	N	N	N	N	Y	Y	Y
Song et al. [25]	N	N	N	N	N	Y	Y	N
Siguang et al. [31]	N	N	N	N	N	Y	Y	N
Verma et al. [35]	N	N	Y	N	N	Y	Y	N
Proposed model	Y	Y	Y	Y	Y	Y	Y	Y

over $Z_{N^2}^*$ and G_1 is set to 160 bits.

In the proposed model, the generation of ciphertext of encrypted value of collected crop-status data ($C_{IDIT_{i,j}}$) needs only one exponential operation in $Z_{N^2}^* - (E_{N^2})$ and one modular multiplication (E_m); also one hash-operation (E_{H_i}) and 2 modular multiplication ($2E_m$) requires to generate the respective tag-value. Then, for local data aggregation at Fog layer, FS_i requires to execute one exponential operation in $Z_{N^2}^* - (E_{N^2})$ and one hash-operation (E_{H_i}) with 2 modular multiplication (E_m); also one modular multiplication (E_m) with one exponential operation (E_{N^2}) requires to verify the validity of collected data as well as authenticity of IoFT devices. Afterward, the leader-node also verifies the validity of aggregated ciphertext with one modular multiplication (E_m) and one exponential operation (E_{N^2}). Then, leader-node generate the transaction (Tx_i) with defined transaction generation cost on blockchain. Finally, to analysis the aggregated data, ACC decrypt the data by parsing the blockchain and decrypting key, which requires only one modular multiplication (E_m) and exponential operation (E_{N^2}). So, the total computational overhead is - $(mw_i|E_{H_i} + E_{N^2} + 3E_m|+4E_m + 2E_{N^2})$. However, in the work proposed by [34], the leader-node requires $(n + 1)$ pairing operation (P) to validate the received transactions, during the global data aggregation. So, due to high computational cost pairing operations, this work requires high computational overhead - $(mw_i|E_{H_i} + (n + 1)P + E_{N^2} + 4E_m|+(n + 1)E_m + 3E_{N^2})$.

Moreover, in the other existing model [31], the local data collection requires 6 exponential operation (E_N) due to providing anonymity to identity of all IoT devices and one hash-operation (E_{H_i}), 3 modular multiplication (E_m). Also, at Fog layer requires $(3E_{N^2} + E_{H_i} + 4E_m)$ operations to verify the authenticity of IoT devices and validity of tag-value. However, due to double layer of blockchain, it requires double number of transactions (Tx_i) generation overhead. However, the decryption overhead at ACC, requires similar computational overhead as the proposed model. So, the total computational overhead of this model is - $(mw_i|E_{H_i} + 6E_{N^2} + 6E_m|+4E_m + 3E_{N^2})$. Furthermore, the compared models [25], [35] requires $(3E_{N^2} + 2E_m + E_{H_i})$ and $(2E_{N^2} + 2E_m + E_{H_i})$ operations for local data aggregation at Fog layer respectively; also requires extra E_m and E_{N^2} operations for global aggregation at Fog layer to verify the authenticity of IoT devices and validity of tag-values. Thus, the final computational overhead of these models are - $(mw_i|E_{H_i} + 4E_{N^2} + 4E_m|+2E_m + 2E_{N^2})$ and $(mw_i|E_{H_i} + 3E_{N^2} + 3E_m|+3E_m + E_{N^2})$ respectively.

Finally, Table III illustrates the detailed comparison of the computational overhead of the proposed model and other

existing models [25], [31], [34], [35]. From Table III, we can observe that the proposed model has lower computational overhead than these compared models [25], [31], [34], [35], due to less number of cryptographic operations. So, the low computational overhead of the proposed model strengthens its practicability.

TABLE III: Comparison of computational overheads

Schemes	Comparison of computational overhead
Siguang et al. [31]	$(mw_i E_{H_i} + 6E_{N^2} + 6E_m +4E_m + 3E_{N^2})$
Lu et al. [34]	$(mw_i E_{H_i} + (n + 1)P + E_{N^2} + 4E_m +(n + 1)E_m + 3E_{N^2})$
Song et al. [25]	$(mw_i E_{H_i} + 4E_{N^2} + 4E_m +2E_m + 2E_{N^2})$
Verma et al. [35]	$(mw_i E_{H_i} + 3E_{N^2} + 3E_m +3E_m + E_{N^2})$
Proposed	$(mw_i E_{H_i} + E_{N^2} + 3E_m +4E_m + 2E_{N^2})$

C. Experimental analysis

For the detailed experimental analysis of the proposed data aggregation model, we performed a set of simulations on the set-up for the designed model. However, in the real-world scenario, some cloud-fog simulators, i.e., iFogSim and YAFS, have been used in research to analyze the network congestion, latency, and cost of cloud-fog environments. Although, these simulators cannot support the implementation of cryptographic operations. Due to this, we have performed a custom implementation of the proposed model. In this set-up, ACC and fog servers are employed on a system with OS Windows 10, Intel(R) Xeon(R) E-2124G CPU@ 3.40GHz \times 64 - based 64 bit processor, 8GB DDR3 RAM, 1TB SATA memory with 64MB buffer. Also, the IoFT devices are simulated with sensor devices. For the sake of simplicity, we simulate an arrangement of 50 fog servers and each fog servers is responsible to collect crop-status data from 20 IoFT devices of corresponding agriculture land. Afterward, fog servers performs the data aggregation and sends the aggregated data to ACC for further analysis.

For a detailed on-chain analysis, we employ the local version of Ethereum's latest client Geth 1.9.0 (such as redactability enabled consortium Ethereum blockchain architecture) to set-up the blockchain framework at Fog layer. Every function of the fog servers and ACC communicates with blockchain via Web3.js API. The Web3.js is a lightweight java library to interact with blockchain through an HTTP connection. To analyze the on-chain computational and communication overhead, we estimate Gas-consumption (in Eth) during each interaction of fog servers and ACC with blockchain.

Without the loss of generality, we assume that number of fog servers changes from 5 to 50 with continuously increasing from 100 IoFT devcies, i.e., 5 fog servers in each test. Fig. 3 illustrates that the computational overhead to locally aggregate

the collected crop-status data by IoFT devices at fog layer for all compared models and proposed model. Fig. 3 also include computational overhead to global aggregation of all collected data with respective transaction (Tx_i) generation over blockchain. From significant observation of Fig. 3, we can see that the proposed model is efficient in terms of computational overhead at IoFT layer, Fog layer, and ACC than any other existing models [25], [31], [34], [35].

For detailed experimental analysis for communication overhead between various layer of the proposed data aggregation model and other existing models, initially, we set the length of IoFT devices - identities (ID_{IT_i}), fog server - identities (ID_{FS_i}), and time-stamp (T_{mt}) be 32 - bits. Then, we analyzed communication overhead, during data transfer from one layer to other layer, i.e., IoFT layer-to-Fog layer and Fog layer-to-ACC by changing number of IoFT devices 5 to 50 in one region. Fig. 4 illustrates the communication overhead of the each compared models [25], [31], [34], [35] with the proposed model. As seen in Fig. 4, the proposed model has lower communication overhead than other existing models to provide more efficient utilization of communication resources.

Furthermore, for concrete experimental analysis, we also illustrate the system performance in terms of latency and throughput of transactions over the blockchain as follows;

1) *Transaction throughput*: refers to the number of successful transactions completed on the blockchain during a specified time slot.

$$\text{Transaction throughput} = \frac{\text{Successful-transactions}}{\text{Time(sec)}}$$

It should be noted that to obtain the successful transactions (Tx_i), the invalid transactions should be removed from the set of total transactions. Fig. 5 illustrates that the average transaction throughput continuously rises as the transaction send-rate rises until it reaches the threshold value of transaction send-rate (tps) 700 transaction-per-second (tsp). Further, in the optimal case, the average transaction throughput is considered to be 700 tsp; after that, the throughput decreases as the throughput decreases as the transaction-send rate is increased.

2) *Transaction latency*: consider the total execution time of a transaction (Tx_i) throughout the blockchain, including the transaction broadcasting time and execution time of the defined consensus algorithm to validate the respective transaction.

Transaction latency = (Transaction confirmation time \times Network threshold) – Transaction submission time

For the sake of simplicity, the system performance for blockchain-assisted proposed data aggregation is obtained by continuously changing the transaction-send-rate (tsp) from 100 to 1000. Fig. 6 shows that the average transaction latency of the proposed model increases considerably until the transaction-send-rate (tps) reaches up-to 690 tps. So, the transaction latency increases linearly as more requests for transaction generation occur; after the transaction-send-rate of 690 tps. So, the average transaction-send-rate of the proposed model is 690 tps.

Therefore, the detailed experimental analysis shows that the proposed data aggregation model is efficient and can be easily implemented in practice.

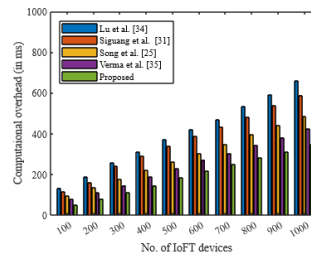


Fig. 3: Comparison of computational overhead

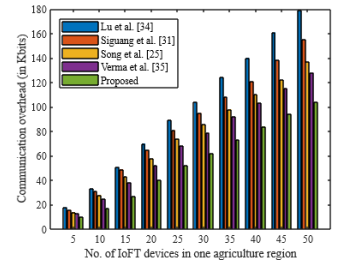


Fig. 4: Comparison of communication cost

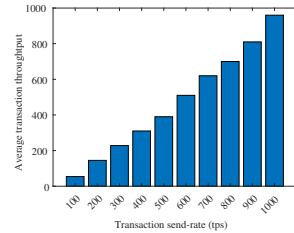


Fig. 5: Transaction throughput of the proposed model

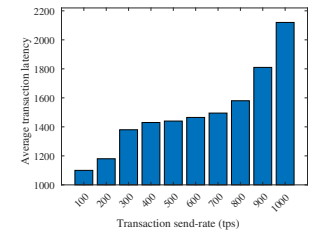


Fig. 6: Average transaction latency of the proposed model

VIII. CONCLUSION AND FUTURE WORK

In this paper, to address the security issues of Fog-enabled smart agriculture paradigm, we introduce an efficient redactable blockchain-based data aggregation model with a secure efficient paillier cryptosystem. The proposed model not only incorporates the fog-computing with smart agriculture but also integrates redactable blockchain to provide reasonable editing to recorded blocks and infeasibility against collusion attacks. The fog layer significantly enhances the model's potential to mitigate the modification attack on aggregated data by integrating the tamper-proof characteristics of blockchain architecture. Further, the detailed concrete security proofs show the computational infeasibility against malicious data mining attacks with proper source authentication, data confidentiality, and integrity under malicious fog servers and external adversaries. Finally, the performance analysis demonstrates the significant advantages, making it a better fit for the real-time data aggregation model for Fog-enabled smart agriculture.

However, the proposed model provides a secure and efficient data aggregation model for IoFT-enabled Fog-paradigm; it still requires a smart consensus algorithm for selecting a leader fog node in a secure decentralized aggregation method. So, in the future, Machine-Learning technology could integrate to increase the practicability of the proposed model in the real world.

REFERENCES

- [1] Goel, R. K., Yadav, C. S., Vishnoi, S., & Rastogi, R. (2021). Smart agriculture—Urgent need of the day in developing countries. *Sustainable Computing: Informatics and Systems*, 30, 100512.
- [2] Malik, A. W., Rahman, A. U., Qayyum, T., & Ravana, S. D. (2020). Leveraging fog computing for sustainable smart farming using distributed simulation. *IEEE Internet of Things Journal*, 7(4), 3300-3309.

- [3] Villa-Henriksen, A., Edwards, G. T., Pesonen, L. A., Green, O., & Sørensen, C. A. G. (2020). Internet of Things in arable farming: Implementation, applications, challenges and potential. *Biosystems engineering*, 191, 60-84.
- [4] Shen, X., Zhu, L., Xu, C., Sharif, K., & Lu, R. (2020). A privacy-preserving data aggregation scheme for dynamic groups in fog computing. *Information Sciences*, 514, 118-130.
- [5] Zhao, S., Li, F., Li, H., Lu, R., Ren, S., Bao, H., ... & Han, S. (2020). Smart and practical privacy-preserving data aggregation for fog-based smart grids. *IEEE Transactions on Information Forensics and Security*, 16, 521-536.
- [6] Guan, Z., Zhang, Y., Wu, L., Wu, J., Li, J., Ma, Y., & Hu, J. (2019). APPA: An anonymous and privacy-preserving data aggregation scheme for fog-enhanced IoT. *Journal of Network and Computer Applications*, 125, 82-92.
- [7] Sarwar, K., Yongchareon, S., Yu, J., & ur Rehman, S. (2021). Lightweight, Divide-and-Conquer privacy-preserving data aggregation in fog computing. *Future Generation Computer Systems*, 119, 188-199.
- [8] Singh, P., Masud, M., Hossain, M. S., & Kaur, A. (2021). Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid. *Computers & Electrical Engineering*, 93, 107209.
- [9] Zhang, L., Li, F., Wang, P., Su, R., & Chi, Z. (2021). A blockchain-assisted massive IoT data collection intelligent framework. *IEEE Internet of Things Journal*.
- [10] Ali, M., Nelson, J., Shea, R., & Freedman, M. J. (2016). Blockstack: A global naming and storage system secured by blockchains. In 2016 USENIX annual technical conference (USENIX ATC 16) (pp. 181-194).
- [11] Ateniese, G., Magri, B., Venturi, D., & Andrade, E. (2017, April). Redactable blockchain—or—rewriting history in bitcoin and friends. In 2017 IEEE European symposium on security and privacy (EuroS&P) (pp. 111-126). IEEE.
- [12] Mishra, R., Ramesh, D., & Mohammad, N. (2022, March). RBDA: Redactable-Blockchain based Secure Data Aggregation Scheme for IoT enabled Cloud Paradigm. In 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (pp. 409-414). IEEE.
- [13] Das, A., & Adhikari, A. (2012). An efficient IND-CCA2 secure Paillier-based cryptosystem. *Information Processing Letters*, 112(22), 885-888.
- [14] Li, X., Xu, J., Yin, L., Lu, Y., Tang, Q., & Zhang, Z. (2021). Escaping from Consensus: Instantly Redactable Blockchain Protocols in Permissionless Setting. *Cryptology ePrint Archive*.
- [15] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16).
- [16] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78, 680-698.
- [17] Lu, R., Heung, K., Lashkari, A. H., & Ghorbani, A. A. (2017). A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT. *IEEE access*, 5, 3302-3312.
- [18] Wang, H., Wang, Z., & Domingo-Ferrer, J. (2018). Anonymous and secure aggregation scheme in fog-based public cloud computing. *Future Generation Computer Systems*, 78, 712-719.
- [19] Lyu, L., Nandakumar, K., Rubinstein, B., Jin, J., Bedo, J., & Palaniswami, M. (2018). PFFA: privacy-preserving fog-enabled aggregation in smart grid. *IEEE Transactions on Industrial Informatics*, 14(8), 3733-3744.
- [20] Saleem, A., Khan, A., Malik, S. U. R., Pervaiz, H., Malik, H., Alam, M., & Jindal, A. (2019). FESDA: Fog-enabled secure data aggregation in smart grid IoT network. *IEEE Internet of Things Journal*, 7(7), 6132-6142.
- [21] Mohammadali, A., & Haghghi, M. S. (2021). A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid. *IEEE Transactions on Smart Grid*, 12(6), 5212-5220.
- [22] Liu, J., Zhao, M., Bao, J., Sun, R., Du, X., & Guizani, M. (2021, December). Fog-Based Conditional Privacy-Preserving Data Batch Verification in Smart Grid. In 2021 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.
- [23] Wang, J., Wu, L., Zeadally, S., Khan, M. K., & He, D. (2021). Privacy-preserving data aggregation against malicious data mining attack for IoT-enabled smart grid. *ACM Transactions on Sensor Networks (TOSN)*, 17(3), 1-25.
- [24] Khan, H. M., Khan, A., Jabeen, F., & Rahman, A. U. (2021). privacy-preserving data aggregation with fault tolerance in fog-enabled smart grids. *Sustainable Cities and Society*, 64, 102522.
- [25] Song, J., Zhong, Q., Wang, W., Su, C., Tan, Z., & Liu, Y. (2020). FPDP: flexible privacy-preserving data publishing scheme for smart agriculture. *IEEE Sensors Journal*, 21(16), 17430-17438.
- [26] Wang, M., He, K., Chen, J., Du, R., Zhang, B., & Li, Z. (2022). PANDA: Lightweight non-interactive privacy-preserving data aggregation for constrained devices. *Future Generation Computer Systems*, 131, 28-42.
- [27] Luo, X., Xue, K., Xu, J., Sun, Q., & Zhang, Y. (2021). Blockchain based secure data aggregation and distributed power dispatching for microgrids. *IEEE Transactions on Smart Grid*, 12(6), 5268-5279.
- [28] Xia, Z., Zhang, Y., Gu, K., Li, X., & Jia, W. (2021). Secure Multi-Dimensional and Multi-Angle Electricity Data Aggregation Scheme for Fog Computing-Based Smart Metering System. *IEEE Transactions on Green Communications and Networking*, 6(1), 313-328.
- [29] Guan, Z., Si, G., Zhang, X., Wu, L., Guizani, N., Du, X., & Ma, Y. (2018). Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. *IEEE Communications Magazine*, 56(7), 82-88.
- [30] Liang, G., Weller, S. R., Luo, F., Zhao, J., & Dong, Z. Y. (2018). Distributed blockchain-based data protection framework for modern power systems against cyber attacks. *IEEE Transactions on Smart Grid*, 10(3), 3162-3173.
- [31] Chen, S., Yang, L., Zhao, C., Varadarajan, V., & Wang, K. (2020). Double-blockchain-assisted secure and anonymous data aggregation for fog-enabled smart grid. *Engineering*.
- [32] Niu, C., Zheng, Z., Wu, F., Tang, S., Gao, X., & Chen, G. (2019). Erato: Trading noisy aggregate statistics over private correlated data. *IEEE Transactions on Knowledge and Data Engineering*, 33(3), 975-990.
- [33] Yan, X., Ng, W. W., Zeng, B., Lin, C., Liu, Y., Lu, L., & Gao, Y. (2021). Verifiable, reliable, and privacy-preserving data aggregation in fog-assisted mobile crowdsensing. *IEEE Internet of Things Journal*, 8(18), 14127-14140.
- [34] Lu, W., Ren, Z., Xu, J., & Chen, S. (2021). Edge blockchain-assisted lightweight privacy-preserving data aggregation for smart grid. *IEEE Transactions on Network and Service Management*, 18(2), 1246-1259.
- [35] Verma, G. K., Gope, P., & Kumar, N. (2021). PF-DA: Pairing Free and Secure Data Aggregation for Energy Internet-Based Smart Meter-to-Grid Communication. *IEEE Transactions on Smart Grid*, 13(3), 2294-2304.
- [36] Yousefi, S., Karimipour, H., & Derakhshan, F. (2021). Data aggregation mechanisms on the internet of things: a systematic literature review. *Internet of Things*, 15, 100427.
- [37] Pourghebleh, B., & Navimipour, N. J. (2017). Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications*, 97, 23-34.
- [38] Zhang, L., Cao, Y., Zhang, G., Huang, Y., & Zheng, C. (2021). A blockchain-based microgrid data disaster backup scheme in edge computing. *Security and Communication Networks*, 2021.
- [39] Boneh, D., Gentry, C., Lynn, B., & Shacham, H. (2003, May). Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 416-432). Springer, Berlin, Heidelberg.
- [40] Jost, C., Lam, H., Maximov, A., & Smeets, B. (2015). Encryption performance improvements of the paillier cryptosystem. *Cryptology ePrint Archive*.
- [41] Rogaway, P., & Shrimpton, T. (2004, February). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *International workshop on fast software encryption* (pp. 371-388). Springer, Berlin, Heidelberg.
- [42] Kanovich, M. I. (1992, June). Horn programming in linear logic is NP-complete. In [1992] *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science* (pp. 200-210). IEEE.
- [43] Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 493-507.