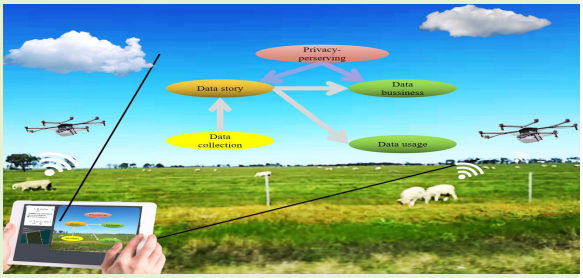
FPDP:Flexible Privacy-preserving Data Publishing Scheme for Smart Agriculture

Jingcheng Song, Qi Zhong, Weizheng Wang, Chunhua Su, Zhiyuan Tan, and Yining Liu✉

***Abstract*— The development of the Internet of Things (IoT) and 5th generation wireless network (5G) is set to push the smart agriculture to the next level since the massive and real-time data can be collected to monitor the status of crops and livestock, logistics management, and other important information. Recently, COVID-19 has attracted more human attention to food safety, which also has a positive impact on smart agriculture market share. However, the security and privacy concern for smart agriculture has become more prominent. Since smart agriculture implies working with large sets of data, which usually sensitive, some are even confidential, and once leakage it can expose user privacy. Meanwhile, considering the data publishing of smart agriculture helps the public or investors to real-timely anticipate risks and benefits, these data are also a public resource. To balance the data publishing and data privacy, in this paper, a privacy-preserving data aggregation scheme with a flexibility property uses ElGamal Cryptosystem is proposed. It is proved to be secure, private, and flexible with the analysis and performance simulation.**



***Index Terms*— Smart Agriculture, Data Aggregation, Data Privacy, Flexibility, Data Publishing.**

I. Introduction

WITH the growing world population, people demand sustainably produced food. Meanwhile, especially due to the COVID-19 pandemic, a growing concern has been paid on food security. For example, many people want to know more about the origin of their products. Smart agriculture can meet these challenges and help farmers to seize growth opportunities. The development of the IoT and 5G technologies have brought changes to smart agriculture. More importantly, boosting the digital agriculture process.

IoT is used greatly to collect agriculture-related information like temperature, humidity, soil PH, soil nutrition levels, water level, animal stress or disease, etc., to inform decisions on irrigation, pest management, fertilizer applications, harvesting, and early detection and treatment of animal disease [1]. These data, which are real-time collected, will be transmitted to Cloud through 5G for easy storage and access [2], [3]. In this way, farmers can use their smartphones to remotely monitor the status of crops or livestock any time and anywhere, and make timely judgements and arrangements. Also, these data can help the government, companies, and academic communities to make some policy and economy decisions or conduct research. Therefore, these data should be regarded as a public social resource.

This work was supported by Natural Science Foundation of China (no. 61662016), Key projects of Guangxi Natural Science Foundation (no. 2018JJD170004), The Science and Technology Major Project of Guangxi Province(no. AA18118039-3), and the study abroad program for graduate student of Guilin University of Electronic Technology(no. GDYX2019003).

However, the agriculture data, same as the personal power messages, personal travel information, and personal health data, often involves users' privacy [4], [5]. For example, these data will expose the income status of farmers. In fact, Cloud can not always be fully trusted, considering the drawbacks and flaws of the current Cloud in terms of privacy-preserving and lack of security support. Therefore, these data cannot be directly shared by Cloud. How to get a trade-off between farmers' privacy and data availability is an urgent problem to be solved [6], [7].

IoT privacy-preserving data aggregation is an essential mean to balance farmers' privacy and data availability, and it is also one of the most important contents for the development of IoT and smart agriculture [8] [9]. Privacy-preserving data aggregation helps the cloud to calculate the sum of specific data, even if Cloud knows nothing about any single data, which makes it as a widely accepted method of protecting privacy. It not only protects farmers' privacy from being violated but also ensures that the data usability can be achieved [10]. Considerable interest has excited in this field in recent years, and various outstanding works have been proposed by researchers.

To solve the privacy and efficiency problems in the smart grid, Li *et al*. in [11] and Lu *et al*. in [12] proposed privacy-preserving data aggregation schemes, respectively. The structure of these solutions can be divided into three parts: the bottom user, the intermediate aggregation center, and the top data center. Specifically, the user encrypted these collected data using homomorphic encryption and sent the results to a

nearby aggregation center. The aggregation center aggregates the ciphertext and sents the result to a remote data center. Since the aggregated data is only $1/n$ in length, these schemes are very effective. Later, Fan *et al*. [13] indicated the above schemes are not secure since they can not resist the attacks from the internal members of the data center. The authors proposed an improved scheme in [13] using a blind factor distributed by an offline trusted authority(TA). The scheme is more secure due to the presence of blind factors, but it is less robust since it requires a TA.

In 2017, Badra *et al*. proposed a new scheme that is secure, efficient, and robust in [14]. It employed the blind factor to resist the attack from internal members. To improve efficiency, the authors adopt a very simple encryption algorithm, but it often leads the scheme to become less secure in fact. Based on previous works, in 2019, Song *et al*. proposed a scheme named DMDA in [15]. In addition to meeting the demands of privacy, security, efficiency, and robustness, the proposal also meets some dynamic requirements. However, it does not satisfy a more advanced dynamic demand. In essence, DMDA only supports farmers' joining and withdrawing. Considering such kind of aggregation scenery, an aggregation system involves four members: Alice, Bob, Carol, and Dave. It is required to aggregate the first data of Alice, the second data of Bob and the third data of Carol, but not Dave's data. To the best of our knowledge, none of the existing schemes can solve this problem.

In this paper, a flexible privacy-preserving data aggregation scheme based on virtual aggregation area, which uses some basic encryption tools such as ElGamal encryption and blind factor, is proposed. With the ingenious use of blinding factors, this proposal supports that the data collector phase and privacy-preserving data aggregation do not need to be done simultaneously. The cloud collects and stores farmers' data in the ciphertext. Then the control center can select some ciphertexts of the data stored in the cloud, and the cloud will calculate the aggregated plaintext and return the results. In these processes, none can get any information about farmers' data.

The analysis shows that the proposed scheme meets the requirements of security, authentication, integrity, privacy, and efficiency. Moreover, some features of the proposed scheme are listed as follow:

Security: it is a basic requirement of cryptographic protocols [16], [17]. The main purpose of security is to ensure that both parties can communicate securely without interruption from the adversary [18], [19].It can be divided into three aspects: confidentiality, authentication and integrity. Confidentiality means that only the specified recipients can access the message. While other unauthorized parties, even if they obtain it somehow, are still unable to understand the content due to the lack of necessary knowledge. Authentication means the message recipients can determine the identity of a legitimate message sender. In other words, the message, which is from the adversary or unknown parties, will not be received or processed. Integrity means the message received by the receiver is the message the sender wants him to get. The message would not be modified or destroyed during transmission. For example, Alice sends AB to Bob, and Bob gets AB rather than A, AC, ABC, or BA. Also, it includes ensuring information non-repudiation and authenticity.

Privacy: it is different from confidentiality. Confidentiality means that the data is secure in the communication, and the adversary cannot get the plaintext even if he gets all the transmitted messages [20], [21]. Privacy here means that the data is secure in all processes of the protocol, and any party, except the data owner, gets nothing about the data even if it is a participant of the protocol [22]. For example, Alice sends a message to Bob and Eve is an eavesdropper. Confidentiality requires Eve to get nothing of Alice's data even if she eavesdrops the message. Privacy meant that not only could Eve not get anything about Alice's data, neither could Bob. However, Bob can leverage Alice's data to some extent [23], [24].

Virtual aggregation area: it is the environment in which the aggregation protocol runs. It includes all participants in the protocol and the communication methods between the participants. The traditional aggregation area usually includes four types of participants: smart devices, aggregators, aggregation centers, and control centers [11]. Participants communicate with each other through a purpose-built network. Generally, the aggregation center of the traditional aggregation area is fully trusted [13], it never adopts proactive attacks and is never defeated. The virtual aggregation area is a bit different. It has three kind of participants: smart devices, Cloud, and control center [14], [15]. The virtual aggregation area, which does not require a special network, is established on the Internet. The communications between participants is conducted via the Internet. The cloud is often unrusted, which leads to higher privacy requirements. Therefore, the proposed scheme requires farmers data to be safe, even if the cloud adopts an active attack or is defeated by an adversary.

Flexibility: it is an advanced feature for a data aggregation scheme. Normally, before the protocol runs, traditional privacy-preserving data aggregation schemes select some farmer, whose data to be aggregated. Then transfer some secure data to these farmer in a secure way. Finally, the encrypted secure data can be aggregated and decrypted [6]. However, flexible solutions are different. A flexible scheme aggregates the collected data rather than collecting data for aggregation [25]. For example, a farmer installs a PH sensor in farm. The PH sensor collects farm PH data in real-time. But this data will be stored in the cloud in ciphertext, taking into account the cost of storage. The farmer can download this data when needed. This means that farmers do not collect and store their own data for aggregation. However, some other organizations, such as company or governments, may also need to use this data. Considering the demand for privacy, aggregation can be adopted to meet the requirements. In the proposed scheme, the control center (CC) generates the aggregation areas according to the requirements and the data stored in the cloud. The cloud can complete the aggregation if all data owners agree. In other words, the purpose of aggregation is to meet the demands of other organizations that use data without compromising farmers' privacy. Therefore, farmers should not be required to collect data for aggregation.

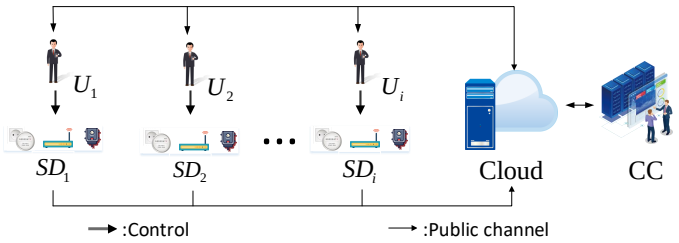


Fig. 1. Systems Model

II. Systems Model

A. Communication Model

The system of this scheme is illustrated in Fig. 3, which consists of four parties: CC, Cloud, user and SDs. We elaborate to introduce them in the following part.

Control Center(CC): CC only takes part in the *Aggregation Phase*, and determines the aggregation space $\mathcal{M}_A$. CC transmits $\mathcal{M}_A$ to the Cloud and receives the result from it. Since the communication between CC and Cloud is not the focus of our research, for simplicity, it is assumed to be secure.

Cloud: Cloud is an important participant of this scheme, which takes part in the *Data Collection Phase* and the *Aggregation Phase*. The Cloud collects data from smart devices and periodically stores it in its database. When the Cloud receives an aggregation request from CC, it aggregates user' data according to the requirements and decrypts the aggregated results with the help of user. Normally, the Cloud is assumed won't to initiate any active attacks, but it tends to know user' privacy by analyzing the legally acquired data. Due to Cloud is a public social resource that anyone can access, the communication between user and Cloud is via a public channel which is easy to be monitored.

Smart Device(SDs): SDs are the data collector owned by user. They collect agriculture data and upload it regularly. In this paper, a farmer can have multiple SDs, for example, a farmer has a smart meter and a smart watch, but an SD can only belong to one farmer. SDs often collect kinds of user' data, encrypt it using user' secure key, and upload it to the Cloud. SDs are considered to have limited computing and storage resources. Therefore, they only attend *Data Collection Phase* that does not require too much calculation.

User: a user is the holder of SDs and data, especially, in smart agriculture, farmer is the user. Users take part in the protocol through smart phone or PCs, so they are assumed to have a little computing and storage resources. Before aggregation, the Cloud needs the support of all data owners. Users can refuse to take part in an aggregation if they think it may destroy their privacy. If one data owner reject this activity, the aggregation cannot continue. In general, most Users are considered trustworthy.

B. Design Goals

This scheme is designed to aggregate data in a virtual area without leaking any single farmer's data. In the big data environment, aggregation is helpful for data analysis and supports management and research of the IoT. To ensure the protocol goes, some important requirements should be met, such as confidentiality, authentication, integrity, and privacy. Moreover, flexibility and efficiency are also important to our scheme.

Confidentiality: Confidentiality means the encryption method is secure. It is impossible to calculate the plaintext from a ciphertext without the security key.

Authentication: Since data transmission is via a public way, the entities receiving the data have to verify the identities of the data senders. Otherwise, the scheme is vulnerable to various attacks, such as the man-in-the-middle attack.

Integrity: Due to the noise pollution in the communication channel or malicious tampering by some adversaries, the information received by the data receivers may be inconsistent with the original one. For ensuring the fake content is not considered to be the correct message received by the data receiver, it should be checked for tampering.

Privacy: Privacy is the most important purpose of a privacy-preserving aggregation scheme. Due to the Cloud is not completely trusted, Cloud should not obtain the plaintext of farmer data.

Flexibility: Flexibility is one of the most important contributions of the proposed scheme. Different from traditional privacy-preserving aggregation methods in physical area, the aggregation schemes in virtual area require higher flexibility. For example, when some analysis tasks require farmers data, the data center can provide some aggregate values. Traditional aggregation schemes can only calculate the aggregation result of a specified aggregation area, which is determined before data collection. Flexibility requires that the aggregation can be achieved without pre-defining the to be aggregated are in advance. Namely, any area is allowed to aggregate.

III. Preliminaries

A. ElGamal Cryptosystem

ElGamal Cryptosystem includes a public key encryption scheme and a signature scheme. They are introduced separately as follows.

1) ElGamal Public Key Encryption: The ElGamal Cryptosystem are based on discrete logarithms, it needs a large prime p, and $p-1$ has at least one large prime factor. Otherwise, computing discrete logarithms is easy (see [26]).

Key Generation: Let p be a large prime and α be a primitive element $mod\ p$ and they are public elements. Secret key x is choosen between 0 and $p-1$, public key is calculated by $y = \alpha^x\ mod\ p$.

Encryption: The plaintext m is encrypted into the ciphertext

$$(c_1, c_2) = (\alpha^r, m + y^r)\ \ mod\ p,$$

where r is a random number between 0 and $p-1$.

Decrption: The ciphertext $\left(c^1, c^2\right)$ is decrypted into plaintext m by

$$m = c^2 - (c^1)^x.$$

TABLE I
NOTATIONS

Acronym	Descriptions
i	The ID number of farmer, such as U_i express the farmer whose ID number is i
SD_i	The smart meter which is controlled by U_i
t	A data tag, such as m_{it} represents the message belonging to U_i and the tag is t
$\mathcal{M}_A$	The aggregation space includes the selected data's related information (i,t)
$\mathcal{U}_A$	The aggregation area consists of the user' ID number i, who have no less than one message $\in \mathcal{M}_A$
x_i/y_i	$U_i's$ secret/public key
x_c/y_c	Cloud's secret/public key
p_i	The decryption piece of U_i

2) ElGamal Signature Scheme: This section introduces a signature scheme about a message m being signed to a pair (s^1, s^2) which satisfy the equation (1):

$$\alpha^m = y^{s^1}(s^1)^{s^2}. \tag{1}$$

The Signing Procedure

In this part, the signature function $SF_x(m)$ will be introduced, which signs the message m using the private key x. The details are as following:

Step 1: Select a fresh random number k which is between 0 and $p-1$, such that $\gcd(k, p-1) = 1 \ mod \ p$.

Step 2: Calculate $s^1 = \alpha^k \ mod \ p$.

Step 3: Calculate $s^2 = \frac{m - xs_1}{k} mod \ (p-1)$, it has a solution for s^2 if k satisfies $\gcd(k, p-1) = 1 \ mod \ p$.

The signature $s = (s^1, s^2)$.

The Verification Procedure

In this part, the verification method is introduced as follows.

Given m, s^1 and s^2, it is easy to verify the signature by checking the equation $\alpha^m = y^{s^1}(s^1)^{s^2}$.

Obviously, if there are more than one signatures such as $\{y_i, m_i, s_i\}$, where $i = 1, 2, 3, \cdots, n$ and $n > 1$, they can be verified together by checking the equation $\alpha^{\Sigma_i \ m_i} = \Pi_i \ y_i^{s_i^1}(s_i^1)^{s_i^2}$.

IV. Flexible Privacy-preserving Aggregation Scheme

In this phase, we introduce a privacy-preserving aggregation scheme based on a random area, which includes three parts: *System Initialization*, *Data Collection* and *Data Aggregation*. The parameters and notations are described in Table I. And an optional data verification is also provided to balance lightweight and security.

A. System Initialization

Before the implementation of the protocol, some parameters should be determined, for example which system parameters generation are selected by Cloud and which farmer parameters are selected by themselves.

1) System Parameter Generation: Let e is a bilinear map $e: G_1 \times G_1 \rightarrow G_T$, G_1 and G_T are cycle groups of order p which p is a large prime and α be a generator of G_1. Moreover, a secure hash function H_1 is selected. $\{e, G_1, G_T, p, \alpha, H_1\}$ is published to all members of system.

2) Farmer Parameter Generation: U_i chooses a secret key x_i and computes a public key $y_i = \alpha^{x_i} \ mod \ p$, which $i \in [1, n]$ is the number of user. Then U_i keeps the x_i and publishes y_i.

Cloud chooses a secret key x_c and computes a public key $y_c = \alpha^{x_c} \ mod \ p$. Then Cloud keeps the x_c and publishes y_c.

B. Data Collection Phase

In this phase, SDs will encrypt, sign and upload its data to Cloud. Moreover, Cloud will verify the received and store the legitimate data. For convenience, we illustrate *data collection phase* by a case: SD_i encrypts his message m_{it} to c_{it}, then it uploads c_{it} to Cloud. More details are introduced as follow:

Step 1: SD_i selects a fresh random number r_{it} between 0 and $p-1$ and calculates $c_{it} = (c_{it}^1, c_{it}^2) = (\alpha^{r_{it}} \ mod \ p, m_{it} + y_i^{r_{it}} \ mod \ p)$.

Step 2: SD_i signs c_{it} to s_{it} which $s_{it} = SF_{x_i}(c_{it}^2)$.

Step 3: SD_i calculates the message authentication code $h_i = H_1(i||c_{it}||s_{it}||T)^{x_i}$ which T is current time, then it uploads $\{c_{it}, s_{it}, h_i, T\}$ to Cloud.

Step 4: When Cloud receives $\{c_{it}, s_{it}, h_i, T\}$, it needs to run the following 3 verification: 1. verifies if T is fresh; 2. verifies if the transmitted message is integrity by $e(h_i, \alpha) = e(H_1(i||c_{it}||s_{it}||T), y_i)$; 3. verifies the signature is correct by the equation $\alpha^{c_{it}^2} = y_i^{s_{it}^1} s_{it}^{1\ s_{it}^2}$. If all yes, Cloud stores (c_{it}, s_{it}).

C. Aggregation Phase

In this phase, Cloud aggregates users' data, which are stored in the Cloud in the ciphertext, and decrypts it with the help of related users. Which data are to be aggregated is determined by CC. Moreover, during the aggression operation, CC, Cloud and the attackers are unable to get any information on user data.

However, Cloud is not always trusted by the users, they may worry Cloud gives a forged aggregated result. Sometimes, users may want to ensure the correctness of the received result. Therefore, an optional authentication is proposed in this scheme. If a farmer is happy to verify the received data, he/she has to bear more complex calculations and inefficient communications. As a return, he/she can enjoy better security due to the verification.

1) Aggregation: In this part, Cloud initially aggregates ciphertexts c_{it} which $(i,t) \in \mathcal{M}_A$ and sends the result to U_i, where $i \in \mathcal{U}_A$. More details are as follow:

Step 1: CC determines an aggregation space $\mathcal{M}_A$ according to demand and sends it to Cloud.

Step 2: Cloud picks out the related ciphertexts c_{it} from its database, where $(i,t) \in \mathcal{M}_A$, and calculates

$$c = \sum_{(i,t) \in \mathcal{M}_A} c_{it}^2 \ mod \ p.$$

Step 3: Cloud calculates the authentication code $h_c = H_1(c||c_{it}^1||\mathcal{M}_A||T')^{x_c}$.

Step 4: Cloud sends $\{c, c_{it}^1, \mathcal{M}_A, T', h_c\}$ to U_i.

2) Initial Decryption: In this part, the user need to complete three steps: 1) verify the received message and decide whether to participate; 2) generate the decryption piece; 3) send the decryption data to Cloud. Especially, in step 1, user not only needs to verify the legality of the received message but also can select to verify if the aggregated result does be the sum of the data in $\mathcal{M}_A$. More details are as follow.

Step 1: After receiving $\{c, c_{it}^1, \mathcal{M}_A, T', h_c\}$, U_i firstly checks the freshness of received messages by T' and verifies its integrity by the equation $e(h_c, \alpha) = e(H_1(c||c_{it}^1||\mathcal{M}_A||T'), y_c)$. Then U_i checks $\mathcal{M}_A$ and decides whether to take part in this aggregation. If all yes, U_i will product the decryption piece. Otherwise, U_i can refuse the aggregation.

Next, U_i verifies if the aggregated result is the sum of the data in $\mathcal{M}_A$. If U_i believes that Cloud is honest, he/she can directly generate the decryption piece for Cloud. If U_i prefers to start the verification, he/she can perform the following processes.

Firstly, U_i asks $\{s_{jt}|(j,t) \in \mathcal{M}_A\}$ from Cloud.

Secondly, U_i verifies the correctness of c by the equation (2)

$$\alpha^c = \prod_{(j,t)\in\mathcal{M}_A} (y_j)^{s_{jt}^1} (s_{jt}^1)^{s_{jt}^2} \ mod\ p. \tag{2}$$

If the equation (2) is working, c passes the verification. Then U_i can calculate the decryption piece.

Step 2: U_i calculates the decryption piece p_i by

$$p_i = c - \sum_{(i,t)\in\mathcal{M}_A} (c_{it}^1)^{x_i} + \sum_{\substack{j\in\mathcal{U}_A \\ j\neq i}} (j-i) y_j^{x_i} \ mod\ p. \tag{3}$$

Step 3: U_i calculates the authentication code $h_i' = H_1(i||p_{it}||T'')^{x_i}$ and sends $\{p_{it}, T'', h_i'\}$ to Cloud.

3) Decryption: When Cloud receives the $\{p_{it}, T'', h_i'\}$, it verifies if the message is fresh and then verifies its integrity by calculating $e(h_i', \alpha) = e(H_1(i||p_{it}||T''), y_i)$.

If Cloud obtains all decryption pieces p_i from user, it can calculate the sum m of $m_{i,t}$, where $(i,t) \in \mathcal{M}_A$. The process is as follow

$$m = c + \sum_{i\in\mathcal{U}_A} (p_i - c) \ mod\ p.$$

V. Security Analysis

A. Confidentiality

The confidentiality of our scheme can divide two parts, one is the confidentiality of data collection phase and another is the confidentiality of aggregation phase. Since the confidentiality of these two parts are completed in different ways, we introduce them respectively:

Confidentiality in Data Collection Phase: Before a data is uploaded, it should be encrypted as $c_{it} = (c_{it}^1, c_{it}^2) = (\alpha^{r_i} \ mod\ p, m_{it} + y_i^{r_i} \ mod\ p)$. Even an adversary eavesdrop c_{it}, he cannot get the plaintext m_{it} from c_{it} due to he cannot get the secure key x_i. According to the [26], no one can calculate plaintext from ciphertext without the secure key. Therefore, the Data Collection Phase can satisfy the requirement of confidentiality.

Confidentiality in Aggregation Phase: In this phase, there are two messages c_{it} and p_i which contain users' privacy data. We have proved the adversary cannot get any information from c_{it}, and then we will prove the adversary also cannot get any information about m_i from p_i. p_i is analyzed as follow.

$$\begin{aligned} p_i =& c - \sum_{(i,t)\in\mathcal{M}_A} c_{it}^{1\,x_i} + \sum_{\substack{j\in\mathcal{U}_A \\ j\neq i}} (j-i) y_j^{x_i} \\ =& \sum_{(i,t)\in\mathcal{M}_A} m_{it} + \sum_{\substack{(j,t)\in\mathcal{M}_A \\ j\neq i}} (m_{jt} + y_j^{r_j}) \\ &+ \sum_{\substack{j\in\mathcal{U}_A \\ j\neq i}} (j-i) y_j^{x_i} \ \mathrm{mod}\ p. \end{aligned} \tag{4}$$

According to (4), if someone wants to calculate m_{it} from p_i, he/she has to know $\sum_{j\in\mathcal{U}_A}^{j\neq i} (j-i) y_j^{x_i} \ mod\ p$. This means he/she knows at least $n-1$ secure keys of the n members of the aggregation area. According to our assumption, most users are trustworthy. Therefore nobody can get so many secure keys. In short, it is impossible to calculate m_{it} from p_i.

B. Authentication

In this section, we will prove that Cloud can ensure the source of the messages in *Data Collection Phase* and Cloud and users can ensure the source of the messages in *Aggregation Phase*, which means the adversary cannot forge a message to pass the authentication.

In *Data Collection Phase*, Cloud will receive the $\{c_{it}, s_{it}, h_i, T\}$ which is from U_i's SD. For ensuring the message does come from U_i's SD, Cloud verifies this message by checking the equation $e(h_i, \alpha) = e(H_1(i||c_{it}||s_{it}||T), y_i)$ with U_i's public key y_i. If an adversary wants to produce h_i, he/she has to obtain the secret key x_i of U_i, which is obviously impossible.

Moreover, in *Aggregation Phase*, Cloud and users authenticate with each other using h_c and h_i', which is very similar to *Data Collection Phase*. Therefore, it is impossible for the adversary to forge messages and pass the authentication.

What's more, sometimes, the user may want to know if the c, which he/she received in step 4 of *Aggregation phase*, dose be the sum of user data in the aggregation area $\mathcal{M}_A$. The authentication is produced with $\{s_{jt}|(j,t) \in \mathcal{M}_A\}$ and the equation (2). According to [26], if $s_{jt} = (s_{jt}^1, s_{jt}^2)$ is the signature of the message m_{jt}, the signature can be verified by calculating $\alpha^{m_{jt}} = (y_j)^{s_{jt}^1} (s_{jt}^1)^{s_{jt}^2}$.

$$left = \prod_{(j,t)\in\mathcal{M}_A} \alpha^{m_{jt}} = \alpha^{\sum_{(j,t)\in\mathcal{M}_A} m_{jt}} = \alpha^c,$$

$$right = \prod_{(j,t)\in\mathcal{M}_A} (y_j)^{s_{jt}^1} (s_{jt}^1)^{s_{jt}^2}.$$

If the equation $left = right$ is working, we can consider that c to be the sum of the user data in aggregation area $\mathcal{M}_A$ unless the adversary can forge a signature of ElGamal, which was proved impossible in [26].

C. Correctness

In *aggregation phase*, U_i can select to verify if c is the sum of the c_{jt} which $(j,t) \in \mathcal{M}_A$. U_i completes this verification by calculating the equation (2). Then we prove the correctness of equation (2):

$$\begin{aligned}\alpha^c &= \alpha^{\sum_{(i,j)\in\mathcal{M}_A} c_{ij}} \\ &= \prod_{(i,j)\in\mathcal{M}_A} \alpha^{c_{ij}} \\ &= \prod_{(i,j)\in\mathcal{M}_A} (y_j)^{s^1_{jt}} (s^1_{jt})^{s^2_{jt}} \ mod\ p.\end{aligned}$$

In fact, $c_{ij} = s^1_{jt}(s^1_{jt})^{s^2_{jt}} \ mod\ p$ is working according to [26].

D. Integrity

In the proposed scheme, the message authentication codes are used to protect the integrity of the transmitted message, such as h_i, h'_i, h_c. We explain the verification equation $e(h_i, \alpha) = e(H_1(i||c_{it}||s_{it}||T), y_i)$ is working as following:

$$\begin{aligned}e(h_i, \alpha) &= e(H_1(i||c_{it}||s_{it}||T)^{x_i}, \alpha) \\ &= e(H_1(i||c_{it}||s_{it}||T), \alpha)^{x_i} \\ &= e(H_1(i||c_{it}||s_{it}||T), \alpha^{x_i}) \\ &= e(H_1(i||c_{it}||s_{it}||T), y_i).\end{aligned}$$

Similar to h_i, the other message authentication messages h'_i, h_c are also working. Moreover, if the message is modified by an adversary, the equation is not working. And the adversary cannot produce a fake h_i since he/she does not have x_i.

E. Privacy

In this subsection, we prove that AC can obtain the sum of m_{it} which $(i,t) \in \mathcal{M}_A$ but it cannot get any information about m_{it}. Firstly, we list all the data available to AC, including m_{it}, as follow:

$$c_{it} = (c^1_{it}, c^2_{it}) = (\alpha^{r_i} \ mod\ p, m_{it} + y_i^{r_i} \ mod\ p),$$

$$p_i = c - \sum_{(i,t)\in\mathcal{M}_A} {c^1_{it}}^{x_i} + \sum_{\substack{j\in\mathcal{U}_A \\ j\neq i}} (j-i) y_j^{x_i} \ mod\ p.$$

Next, we explain that AC cannot calculate m_{it} using c_{it}, p_i, nor can it obtain m_{it} from p_i.

If AC wants to decrypt c_{it}, AC has to get the $U'_i s$ privacy key x_i or crack the ElGamal cryptosystem. However, x_i is produced by U_i and kept secret, and ElGamal cryptosystem is proved secure according to [26]. Therefore these two solutions are infeasible.

AC cannot obtain m_{it} from p_i since $\sum_{j\in\mathcal{U}_A}^{j\neq i}(j-i)y_j^{x_i}$ is difficult to calculate without knowing x_i. Due to x_i is a secret only known by U_i, it is impossible to calculate m_{it} from p_i.

When a single farmer is required to take part in an aggregation process, but if he/she is worried that the aggregation may lead to privacy leakage, he/she can refuse to provide decryption piece. If someone in aggregation area does this, Cloud cannot complete the aggregation even if all other members agree to participate in the aggregation.

TABLE II
SECURITY COMPARISON WITH OTHER SCHEMES

	Li *et al.*	Fan *et al.*	Badra *et al.*	DMDA	Proposed scheme
Confident	Yes	Yes	Yes	Yes	Yes
Authentication	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes
Inside attack	No	Yes	Yes	Yes	Yes
Decentralize	No	No	Yes	Yes	Yes
Dynamic	No	No	No	Yes	Yes
Flexibility	No	No	No	No	Yes

F. Flexibility

When CC produces the aggregation area $\mathcal{M}_A$, CC can choose any data stored in Cloud. No matter which aggregation area $\mathcal{M}_A$ is selected, Cloud can calculate the aggregated result with the help of related users. In fact, CC can select several data that are owned by some different users, such as $\mathcal{M}_A = \{(1,t_1),(2,t_2),(3,t_3)\}$, or owned by only one farmer, such as $\mathcal{M}_A = \{(1,t_1),(1,t_2),(1,t_3)\}$, or mixed, such as $\mathcal{M}_A = \{(1,t_1),(2,t_2),(2,t_3)\}$.

Not all users take part in a data aggregation task, a farmer is considered to take part in the process only when at least one data of him/her is in the aggregation area.

G. Source Authentication

In the proposed scheme, it is free for users to decide whether to verify if the aggregated result or not. In this subsection, we prove if users want to verify the aggregated result, then they can do it, while Cloud cannot deceive them.

According to the subsection *Initial Decryption*, users verify the source by the equation as follows:

$$\alpha^c = \prod_{(j,t)\in\mathcal{M}_A} (y_j)^{s^1_{jt}} (s^1_{jt})^{s^2_{jt}} \ mod\ p.$$

In this equation, we can find that if Cloud wants to forge fake data to pass this verification, he has to produce the fake signature (s^1_{jt}, s^2_{jt}) to users. However he cannot achieve this since he does not have the private key of the connected user.

Moreover, in order to better demonstrate the superiority of our article, the proposed scheme is compared with some related works as Table II. By comparison, the proposed scheme is more advantageous in terms of flexibility and dynamics

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed scheme in terms of the communication cost, storage cost as well as calculation cost.

TABLE III
MESSAGES LENGTH

Message Type	Length
Plaintext	64 $bits$ to 128 $bits$
Ciphertext	128 $bits$ to 256 $bits$
Hash result	128 $bits$
Signature	128 bit to 256 $bits$
Time	64 $bits$
Label	64 $bits$
Secure key	128 $bits$
Public key	128 $bits$

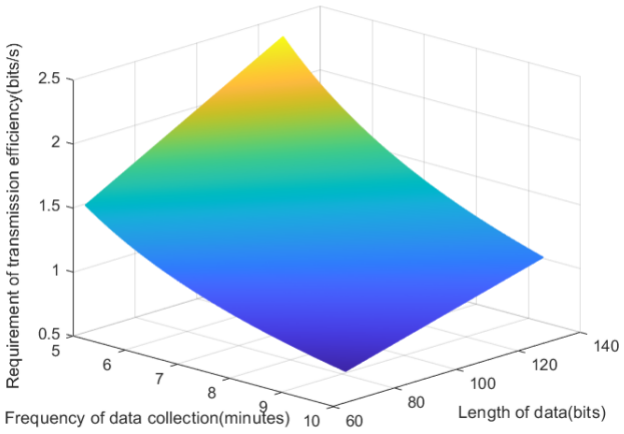


Fig. 2. The relationship between transmission efficiency, the frequency of data collection, and the length of the collected data

A. Communication Cost

In the proposed scheme, there are four communication modes: the communication between SD and Cloud, the communication between SD and user, the communication between users and Cloud, and the communication between Cloud and CC. Users can only control their SDs physically, while the communication between Cloud and CC is not within the research category of this paper. Therefore, two kinds of communication between user and SD and between Cloud and CC are not be discussed here. For simplicity, the length of various messages is listed in Table III.

The communication between SD and Cloud appears in *Data Collection Phase*, which SD sends the collected data $\{c_{it}, s_{it}, h_i, T\}$ to Cloud. Because data collection is independent of aggregation, the communication cost between SD and Cloud is only related to the frequency of data collection and the length of the collected data. Normally, in smart agriculture, the length of the collected data is not very long, since the collected data is always a PH value or a temperature value. Moreover, users do not need collect data very frequently. Therefore, we assume the collected data is between 64 $bits$ to 128 $bits$, and data is collected every 5 $minutes$ to 10 $minutes$. According to the above assumption, the relationship between transmission efficiency, the data acquisition frequency, and the length of the collected data is obtained, as shown in Fig 2.

According to Fig 2, the communication efficiency of smart devices is required to be bigger than 2.5 $bits/s$, which is easy to meet in fact. Sometimes, data needs to be collected more frequently. But even the data is upload once every 5 seconds, the communication efficiency requirement is smaller than 150 $bits/s$, which is also acceptable.

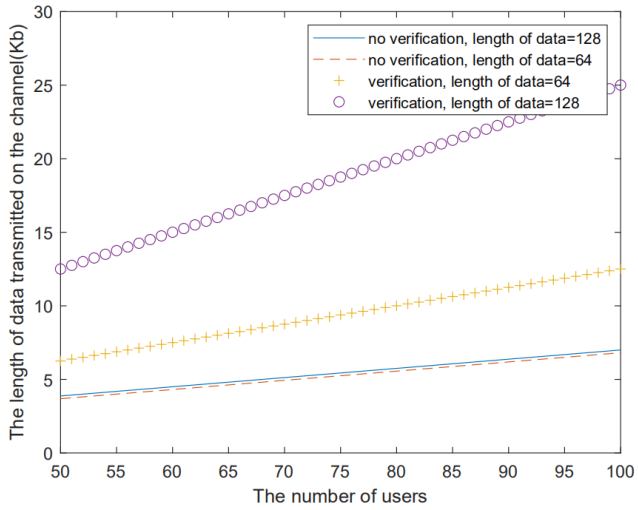


Fig. 3. The relationship between the length of data transmitted in channel, the number of users, the and the length of data

Moreover, the communication between users and Cloud appears in *Aggregation Phase*. In this phase, the Cloud sends the aggregation data $\{c, c_{it}^1, \mathcal{M}_A, T', h_c\}$ to user, and the user chooses to verify $\mathcal{M}_A$, then he/she returns an application information to Cloud, and Cloud sends $\{s_{jt}|(j,t) \in \mathcal{M}_A\}$ back to user. Finally, user sends the decryption information $\{p_{it}, T'', h_i'\}$. Therefore, the communication cost is related to the number of users in the aggregation area and the length of data. In smart agriculture, for balancing the privacy and the data availability, the number of users is assumed to be between 50 to 100. Then we can get the relationship between the length of data transmitted in the channel, the number of users, and the length of data, as shown in Fig 3.

According to Fig 3, there are 4 lines here representing the communication consumption when the user does not verify $\mathcal{M}_A$ and the length of data is 128 bits, the user does not verify $\mathcal{M}_A$ and the length of data is 64 bits, the user verifies $\mathcal{M}_A$ and the length of data is 64 bits, and he user verify $\mathcal{M}_A$ and the length of data is 128 bits. When the user does not verify $\mathcal{M}_A$, the communication cost is very low, i.e. less than 7 Kb. However, if a user wants to verify $\mathcal{M}_A$, the communication cost will increase a lot, but still not exceed 25 Kb. According to the assumption, users can use their smartphones or laptops to participate in the protocol. Therefore, the communication cost, which is less than 25 Kb, is acceptable.

B. Storage Cost

Users, especially their SDs, are most concerned about the sensitivity of storage cost. In this subsection, the storage cost of users and SDs are listed and compared with other related schemes.

A SD only needs to store a private key and a public key owned by U_i. Therefore, the storage cost of a smart device is only 128 bit + 128 bit = 256 bit. Although SD only has a limited storage resource, 256 $bits$ is acceptable.

TABLE IV
NOTATIONS

Notation	Descriptions
T_{add}	Cost of an addition
T_{mul}	Cost of a multiplication
T_{exp}	Cost of an exponential operation
T_{bil}	Cost of a bilinear mapping operation
T_H	Cost of a hash operation

TABLE V
COMPUTATION OVERHEAD

Entity	T_{add}	T_{mul}	T_{exp}	T_{bil}	T_H
SD	2	2	4	0	1
U_i without verification	$2n+t_i$	$n+1$	$n+t_i+1$	2	1
U_i with verification	$2n+t_i$	$\sum_n t_i+$ $n+1$	$2\sum_n t_i$ + $n+t_i+1$	2	1

U_i needs to maintain a public key table containing all users' identification numbers and public keys, therefore the storage cost of a farmer is $n \times (16\ bits + 256\ bits) + 256\ bits = (272n + 256)\ bits$ which n is the number of members.

According to the assumptions, Cloud has enough storage resources, therefore we are not discussing the storage costs of Cloud here.

C. Calculation Cost

Calculation cost is an important factor in evaluating a security scheme. Normally, if an entity has insufficient computing resources, it should undertake little calculation. In the proposal, SDs have the least calculation resources and the users have the second least calculation resources. Therefore, the calculation cost of SDs and the users are discussed below. For simplicity, some notations are defined in Table IV. The performance evaluation is executed on a laptop with the Intel Core i7-7700HQ CPU @2.8GHz and 8GB memory, which is based on the PBC and Openssl libraries.

1) SDs Calculation cost: SDs only take part in *Data Collection Phase*, they encrypt plaintext m_{it} to ciphertext c_{it}, sign the c_{it} using the ElGamal cryptosystem, and produce a message authentication code using a hash function H_1. The encryption and signature processes is $(c_1, c_2) = (\alpha^r, m + y^r)\ mod\ p$, $\left(s^1, s^2\right) = \left(\alpha^k, \frac{m - xs_1}{k}\ mod\ (p-1)\right)$. According to the above formulas, the cost of encryption is $T_{add} + 2T_{exp}$ and the cost of signature is $T_{add} + 2T_{mul} + T_{exp}$. Moreover the message authentication code costs $T_{exp} + T_H$. Therefore SD will cost $2T_{add} + 2T_{mul} + 4T_{exp} + T_H$ for once data collection as shown in Table V. Normally, the very few calculation costs are ignored. A data collection, including an encryption of plaintext and a message authentication code, only takes about $5ms$. For any SD, the cost is easy to bear.

2) Users Calculation cost: Users use their laptop or smartphone to take part in the protocol, therefore they have limited calculation resources. The user U_i works in *Data Aggregation Phase* to initially decrypt the aggregation result. He/she verifies the message authentication code using $e(h_c, \alpha) = e(H_1(c||c_{it}^1||\mathcal{M}_A||T'), y_c)$, calculates the decryption piece using the formula (3), and calculates the message authentication code using $h_i' = H_1(i||p_{it}||T'')^{x_i}$. According to above formulas, the cost of verification is $T_{mul} + 2T_{bil}$, the cost of decryption piece production is $(2n + t_i)\, T_{add} + nT_{mul} + (n + t_i)\, T_{exp}$, where t_i is the data number of U_i participating in this aggregation, and n is the number of users participating in the aggregation. Each user should cost $T_{exp} + T_H$ to generate the authentication code. Note: the amount of data for each user participating in the aggregation processes may vary. Moreover, if U_i decides to verify c, he/she needs $\left(\sum_n T_i\right) T_{mul} + 2\left(\sum_n T_i\right) T_{exp}$ extra cost. Therefore, if U_i does not verify c, the cost of an aggregation is $(2n + t_i)\, T_{add} + (1 + n)\, T_{mul} + (n + t_i + 1)\, T_{exp} + 2T_{bil} + T_H$; otherwise, he/she will cost $(2n + t_i)\, T_{add} + (1 + n + \sum_n t_i)\, T_{mul} + (n + t_i + 1 + 2\left(\sum_n t_i\right))\, T_{exp} + 2T_{bil} + T_H$, as listed in Table V. When $n = 100$ and $t_i \equiv 1$, in no verification case, the user will cost $162ms$, otherwise, he/she will cost $422ms$. In fact, if $t_i 1$, when the number of aggregated data $\sum_n t_i$ is also 100, the cost will be lower. Due to users take part in the protocol by a smartphone or laptop, the calculation cost is easily withstanding.

VII. CONCLUSION

In this paper, we proposed a flexible privacy-preserving data aggregation scheme base on smart agriculture, which supports optional data aggregation in the virtual aggregation area. The analysis shows that the proposed scheme is secure, privacy-preserving and efficient. In the future, we plan to improve the degree of decentralization. For example, replacing Cloud with Block-chain.

REFERENCES

[1] K. A. Patil and N. R. Kale, "A model for smart agriculture using IoT," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*., pp. 543-545, Dec. 2016.

[2] D.V. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare informatics research*, vol. 22, no. 3, pp. 156-163, 2016.

[3] Y. Sun and H. Song and A. J. Jara and R. Bie, "Internet of Things and Big Data Analytics for Smart and Connected Communities," *IEEE Access*, vol. 4, pp. 766-773, 2016.

[4] M. E. Sykuta, "Big Data in Agriculture: Property Rights, Privacy and Competition in Ag Data Services," *International Food and Agribusiness Management Review*, no. 1030-2016-83141 pp. 18, Jun. 2016.

[5] A. Dorri and S. S. Kanhere and R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 618-623, March 2017.

[6] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.

[7] Y. Liu, C. Cheng, and T. Gu, T. Jiang, and X. Li, "A lightweight authenticated communication scheme for smart grid," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 836–842, 2016.

[8] J. Y. Ferris, "Data privacy and protection in the agriculture industry: is federal regulation necessary," *Minn. JL Sci. & Tech.*, vol. 18, pp. 309, 2017.

[9] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu, "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82-92, 2019.

[10] Z. Lv, H. Song, P. Basanta-Val, A. Steed and M. Jo, "Next-Generation Big Data Analytics: State of the Art, Challenges, and Future Research Topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891-1899, Aug. 2017.

[11] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *International Journal of Security and Networks*, vol. 6, no. 1, pp. 28-39, 2011.
[12] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.
[13] C.-I Fan, S.-Y Huang, and Y.-L Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 666–675, 2014.
[14] M. Badra, S. Zeadally, "Lightweight and efficient privacy-preserving data aggregation approach for the Smart Grid," *Ad Hoc Networks*, vol. 64, pp. 32-40, 2017.
[15] J. Song and Y. Liu and J. Shao and C. Tang, "A Dynamic Membership Data Aggregation (DMDA) Protocol for Smart Grid," *IEEE Systems Journal*, pp. 1-9, 2019, 10.1109/JSYST.2019.2912415.
[16] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security & Privacy*, no.5 pp. 75–77, 2009.
[17] I. Butun, P. Österberg and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616-644, Firstquarter 2020.
[18] F. Aloul, AR Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj, "Smart grid security: Threats, vulnerabilities and solutions," *International Journal of Smart Grid and Clean Energy*, vol.1 no.1 pp. 1–6, 2012.
[19] H. Song, R. Srinivasan, T. Sookoor, and S. Jeschke, "Smart Cities: Foundations, Principles and Applications." *NJ: Wiley*, ISBN: 978-1-119-22639-0, Hoboken, pp.1-906, 2017.
[20] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pp. 61–66, 2010.
[21] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang and X. Yao, "Security and Privacy Preservation Scheme of Face Identification and Resolution Framework Using Fog Computing in Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1143-1155, Oct. 2017.
[22] G. Kalogridis, C. Efthymiou, S. Z Denic, T. A Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 232–237, 2010.
[23] H. Song, G. A. Fink, and S. Jeschke, "Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications," *UK:Wiley-IEEE Press* ISBN: 978-1-119-22604-8, Chichester, pp. 1-472, 2017.
[24] C. Lin, Z. Song, H. Song, Y. Zhou, Y. Wang, and G. Wu. "Differential privacy preserving in big data analytics for connected health." *Journal of medical systems* vol. 40, no. 4 pp. 97, 2016.
[25] W. Tang and J. Ren and K. Deng and Y. Zhang, "Secure Data Aggregation of Lightweight E-Healthcare IoT Devices With Fair Incentives," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8714–8726, 2019.
[26] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Advances in Cryptology*, pp. 10–18, 1985.