Hindawi
Security and Communication Networks
Volume 2021, Article ID 6668339, 18 pages
https://doi.org/10.1155/2021/6668339

Research Article

Cross-Department Secures Data Sharing in Food Industry via Blockchain-Cloud Fusion Scheme

Q. Tao,[1] **Q. Chen**,[2] **H. Ding**,[1] **I. Adnan**,[1] **X. Huang**,[3] **and X. Cui**[1]

[1]*Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China*
[2]*School of Remote Sensing and Information, Wuhan University, Wuhan 430072, China*
[3]*School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621002, China*

Correspondence should be addressed to X. Cui; xcui@whu.edu.cn

Received 21 December 2020; Revised 23 January 2021; Accepted 27 February 2021; Published 27 March 2021

Academic Editor: Leandros Maglaras

The barriers of food enterprises and departments caused information asymmetry, which is the root cause of food safety incidents. Simultaneously, it is challenging to solve the information asymmetry by the existing cloud-based food supply-chain regulation system. Establishing a secure and reliable data sharing environment is an effective solution to the information island. Blockchain can construct a security network based on mathematical algorithms, eliminating the third party's potential security risk, and realize transparently share data. In this paper, on the principle of metadata remaining in the food enterprises, we propose a blockchain-cloud fusion scheme based on Decentralized Attribute-Based Signature (DABS) to realize secure data sharing between departments. It constructs a decentralized and trusting environment for data owners to share data and achieves social co-governance of food safety based on the smart contract. It can also preserve the existing system architecture and complement the performance disadvantage of blockchain and cloud storage. The result achieved from security analysis shows that our scheme supports unconditional full anonymity and can resist collusion attacks of N-1 out of N corrupted attribute authorities.

1. Introduction

In the age of big data, the data has a tremendous potential value, and different enterprises and regulators have collected data, such as farmers can only collect the data of crop growth (*seeds, fertilizers, pesticides, environment, etc.*). Food producers can get the processing data (*product formulation, machine, process parameters, inputs, etc.*), sellers collect sales data (*location, price, customer, etc.*), and regulators store the sampling inspection reports and monitoring data. The data owners would not share the data with the disclosure risk of user privacy and commercial secret. It is the primary cause of information asymmetry that has led to some negative market phenomena, like "good money after bad," "2013 horse meat scandal in the Europe," and "2017 multistate *Salmonella* outbreak in the US" [1]. More than 600 million people worldwide fall ill after consuming unsafe food each year [2]. Cross-departmental data sharing in the food industry is a promising solution to food safety incidents and promotes industry development [3]. The traditional cloud-based regulatory system (as Figure 1 shown) provides a solution to share the food data and protect the food quality [4]. According to the Cisco white paper, most companies in the food supply-chain deployed the regulatory system in cloud [5]. Cloud features of pay-on-demand and elastic extension can decrease the cost [6–8]. However, the customers and data in the cloud are not in the same trusted domain, resulting in a lack of trust between customers. Cloud security incidents are frequent. More than 10000 security incidents happened in Malaysia in 2013. According to the 2013 Norton report, the total cost of cybercrime in Australia amounts to AU$1.06 billion [9]. The privacy of 368000 students in *Florida Virtual School* was leaked in 2018. So, some common defects in the traditional regulatory system need to be addressed: (1) the tampering and hiding of wrong information in the centralization system. (2) the risk of privacy

leakage and data loss if the servers are compromised or privileged users' rights are not adequately monitored. (3) It is difficult to verify the user's identity, which is hard to guarantee the authenticity of shared data and track down the person responsible for the product accident. (4) The data owners are reluctant to share data due to the lack of trust between the system and food enterprises, and the enterprises would not invest heavily to rebuild their system. More than 40% viewed that food fraud is difficult to monitor by the traditional methods [2].

With the characteristics of tamper-proofing, decentralization, and co-governance, blockchain happens to address the above issues by constructing a trusted network based on a mathematical algorithm. The data are shared transparently by enterprises and regulators. It is immutably for the data in block chain [10]. Smart contract is an intelligent and self-executing logic code without an intermediary, reducing transaction costs and transaction time [11]. *Walmart* has developed a blockchain-based system to monitor the product quality by sharing information in pork and mangos industries [12]. It helps to effectively track the pork products during serval minutes compared to serval days taken in the past. There are several challenges: (1) the blockchain has the characteristic of pseudoanonymity that cannot protect the privacy information in the signature. The adversary can get the privacy information by analyzing the expenditure information of user account [13]. (2) Performance, cost, and security are the primary bottlenecks for implementing blockchain technology in the food industry.

To address the issues, we propose a blockchain-cloud fusion scheme to protect the security of shared data. The enterprises can preserve the existing system architecture and transparently share data in a trusted network without third parties. It adopts the characteristics of low cost, scalability, and high-performance in cloud computing technology to make up for blockchain's performance and cost bottleneck. The metadata remains in the existing enterprise system, and the data's signature is shared with the blockchain network. It can reduce the storage and performance load of the blockchain network. Besides, the analysis of the signature algorithm in Table 1 shows that the Attribute-Based Signature (ABS) is an effective solution to share data with fine-grained control and protect the data owner's privacy. To address the information leakage risk and adapt to the decentralization feature of our scheme, we propose a Decentralized Attribute-Based Signature (DABS). The enterprises and regulators have equal rights to verify employees in their respective departments. The signature of shared data is entirely secure, which encourages users to monitor food quality actively. Simultaneously, it provides a solution for regulators to track down the person responsible for public safety incidents and rumors. It is meant to promote social co-governance in the food industry.

1.1. Related Work

1.1.1. ABS. ABS originated from fuzzy identifies encryption that was firstly proposed by *Sahai and Waters* [14]. It could hide user identity information and provide a solution for data owners to share with fine-grained access control. In the ABS, the users received a private key from the Attribute Authority (AA) based on their attributes and defined a shared community for sharing data with a signing predicate. Only if the users' attributes satisfy with the signing predicate, they could get the share data, such as farmers share a file with an access control strategy of signing predicate ((*manager with Level 7 in Food Processing FP1*) *or regulator*), which means that only *the manager with Level 7 in Food Processing FP1* and *regulators* can access the shared file. ABS scheme has a bright application prospect, such as directional broadcast and cloud storage. It has attracted many scholars and presented a lot of research results [15]. Li et al. proposed an Attribute-Based Encryption (ABE) system based on a ring signature scheme [16]. Anyone can select a set of public keys of random signers to hide its public key [17]. However, the above schemes manage the entire system attribute set by a single AA. It quickly causes performance bottlenecks and cannot satisfy the actual needs of multiple departments' cooperation [18]. So, Chase et al. let numerous authorities manage the attribute set, but the scheme has disclosure risk of Centralized Attribute Authority (CA) because the CA could calculate any user's private key [19]. Yang et al. proposed an efficient multiauthority CP-ABE scheme based on LSSS access structure without global authority and security under the random oracle model [20]. Liu et al. proposed a security scheme only if the number of colluding users is less than $(m+1)$ [21].

1.1.2. Blockchain. *Nakamoto* firstly proposed the blockchain architecture [10]. It provided a solution to a trust and equality among different participators by a mathematical algorithm. In addition, it is an effective way to solve the single point validation based on decentralization [22]. Macrinici et al. [11] designed the smart contract to protect users' privacy and automatic information processing. Blockchain technology has achieved great success in the financial industry [23], like *Bitcoin* and *Ethereum.* It demonstrated the application feasibility of blockchain technology. Some promising blockchain applications are being developed to address industry concerns such as medical, agriculture, energy, and food safety. In [24–28], a security data management scheme was proposed with privacy-preserving to share the medical data by encapsulating EHRs based on attribute-based encryption into the blockchain; there is no detailed application. Even in [28], it cannot resist the collusion attack between uses. In [29], it is proposed that blockchain is a solution to optimize the energy industry structure and facilitate sustainable development. *Walmart* developed a blockchain-based system to monitor pork and mangos from South America to the US [1], where the managers could trace down the product during several minutes compared to several days taken in the past.

1.1.3. Supply Chain and Blockchain. Data-shared barriers among the food enterprises caused information delay and

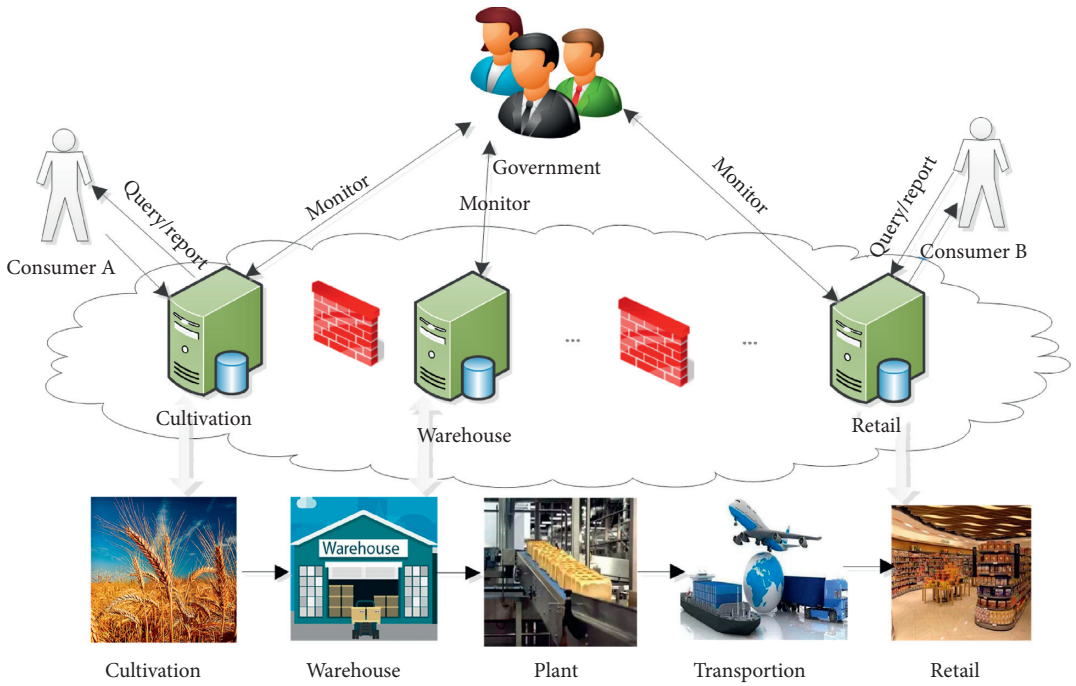

Figure 1: Traditional food supply-chain regulation system (the barriers among enterprises and departments block information sharing in the food supply chain).

Table 1: Signature algorithm analysis.

Algorithm	Features	Main application
RSA signature	Algorithm security is based on large prime decomposition. The private key space is large, the encryption effect is good, but the encryption speed is slow, and it is very complex to the management of the public key authentication and certificate. 1 : 1 communication mode.	Short message signature
Blind signature	The message is blind to the signer. Even the signature is leaked by the recipient; it cannot be tracked by the signer; 1 : 1 communication mode.	Electronic cash; electronic election Pay-tv
Broadcast signature	Transmit the signed messages to a group of users with an insecure channel;. But, it is difficult to obtain the size and membership of the receiving group in time under the open mode; enumerate user identities in distributed applications can compromise user privacy.	Video conference
Group signature	The anonymous signature is used for group sharing, and the receiver can verify the signature but cannot get the signer's information.	Electronic bidding
	The group administrator creates the group with key distribution, which has the risk of identity escrow leakage and signature forging.	Digital copyright protection
Attribute-based signature (ABS)	Signature and user private key associated with attributes.	Fine-grained access control; group key management
	Fine-grained noninteractive access control and 1 : n communication mode; it reduces the network bandwidth for shared data and the processing overhead for nodes. Resist to collusive attacks, anonymous.	Privacy protection

asymmetry, which affected the quality of shared information [30]. Supply chain management is an important application for blockchain technology [31]. Blockchain could record entire life cycle of each product with immutable and shared information between consumers and producers. The ICT electronic agriculture system via blockchain infrastructure guaranteed the integrity of agricultural environment data, conducive to improving sustainable agrarian development [32]. In [33], it is proposed that blockchain applied in the food supply-chain not only could reduce food losses by optimizing product logistics, but it also benefits to improve regulatory efficiency. *Saberi et al.* explored how blockchain could help supply-chain sustainability and guide industry transformation. *Clauson and Breeden* discussed the supply-chain management in healthcare, and most blockchain schemes are still in the

proof-of-concept or pilot stag [34]. Security and privacy are barriers to the integration of IoT and blockchain. *Jangirala* et al. proposed a LBRAPS protocol in mobile edge computing to protect the transmit data security [35]. In [36, 37], a AgriBlockIoT system is proposed for food supply-chain management. But, it lacks an effective solution to protect the data security.

1.2. Our Contributions. In summary, main contributions of this work are as follows:

(1) This work proposes a food supply-chain regulation system based on a blockchain-cloud fusion scheme. It supports secure data sharing across departments, intelligent regulation, and social co-governance with a smart contract.

(2) This work proposes a DABS scheme by combining the characteristics of ABS and ring signature. It helps to improve the safety of the food supply-chain regulation system. Besides, it provides a solution for regulators to trace down the illegal users, which is useful to prevent the spread of rumors and establish a harmonious network environment.

(3) This work performs a comprehensive security analysis shows that our scheme supports unconditional full anonymity and noncollusion with strong (N-1) (resist collusion attacks of N-1 out of N corrupted Attribute Authorities). The performance evaluation shows our scheme's performance advantage. Compared with time complexity $nO(\cdot)$ in traditional schemes, it supports batch-verification with $O(\cdot) + n$.

1.3. Organization. The remainder of this work is as follows: Section 2 introduces the definitions of the bilinear map, computational assumption, access structure and LSSS, syntax, and security model of DABS. Section 3 details the blockchain-cloud fusion scheme. Section 4 describes security analysis, and Section 5 presents the performance analysis, while the conclusion is presented in Section 6.

2. Preliminaries

This section introduces the definitions of bilinear map, computational assumption and linear secret-sharing schemes. Then, we describe the framework definition of the DABS scheme and security definitions.

2.1. Bilinear Map. Let G_1, G_2, and G_T be three multiplicative groups of prime order p. A bilinear map is a map e: $G_1 \times G_2 \longrightarrow G_T$ with the following properties:

(1) Bilinearity: $e(f^a, h^b) = e(f, h)^{ab}$ for $\forall a, b \in Z_p \forall f \in G_1, \forall h \in G_2$

(2) Nondegeneracy: $e(f, h) \neq 1$

(3) Computability: there is an efficient algorithm to compute $e(f, h)$ where $\forall f \in G_1, \forall h \in G_2$

If $\forall f \in G_1, \forall h \in G_2$, then $e(f, h) \in G_T$. The bilinear pairing applied in our proposed scheme is symmetric, where $G_1 = G_2 = G$.

2.2. Computational Assumption

Definition 1. Computational Diffie-Hellman problem, CDH: assume G_1 is a bilinear group of prime order p. g is a generator of G_1, bilinear map e: $G_1 * G_1 \longrightarrow G_T$. Giving (g, g^a, g^b) for unknown $a, b \in Z_p$ to compute g^{ab}. We say that the (t, ε)-CDH problem holds if there exists no poly(t)-time algorithm can solve the CDH problem with non-negligible advantagesε.

2.3. Access Structure and Linear Secret-Sharing Schemes (LSSS)

Definition 2. Access structure [38]: let $U = \{U_1, U_2, U_3, \ldots, U_n\}$ be a set of parties, and a collection $A \subseteq 2^{\{U_1, U_2, U_3, \ldots, U_n\}}$ is called monotone if $\forall B, C \in 2^{\{U_1, U_2, U_3, \ldots, U_n\}}$: if $B \in A$ and $B \subseteq C$, then $C \in A$. A monotone access structure is collection A which is a non-empty subsets of $\{U_1, U_2, U_3, \ldots, U_n\}$, i.e., $A \in 2^{\{U_1, U_2, U_3, \ldots, U_n\}} \setminus \{\phi\}$. The set in A is called authorized set, and others is an unauthorized set.

Definition 3. Linear secret sharing scheme [39] (LSSS): a secret sharing scheme $\prod$ over a set of parties $U = \{U_1, U_2, U_3, \ldots, U_n\}$ is LSSS only if

(1) The share for each party forms a vector over Z_p.

(2) There exists a share generating matrix M with l rows and m columns. For $i = 1, \ldots l$, let the function ρ map the row i of M to the attribute $\rho(i)$. When we consider the column vector $\overrightarrow{v} = (v_1, v_2, \ldots, v_m)^T$, where $v_1 = s \in Z_p$ is the secret to be shared, and $v_2, \ldots, v_m \in Z_p$ are chosen randomly; then $M\overrightarrow{v}$ is the vector which shares the secret s by function $\prod$. The shared secret $\lambda_i = M_i \cdot v$ belongs to the party $\rho(i)$.

Suppose that $\prod$ is an LSSS for the access control strategy γ. Let $A \in \gamma$ is an authorized set, and $I \subseteq \{1, 2, \ldots, l\}$ is defined as $I = \{i: \rho(i) \in A\}$. If λ_i is valid shares of any secret s by function $\prod$, there exists constant $\omega_i \in Z_p$ and $\sum_{x \in \rho_{(S_i)}} (\omega_i * \lambda_i) = s$. In [38], it is shown that these constants $\{\omega_i\}_{i \in I}$ can be found in polynomial time.

2.4. Syntax of Decentralized Attribute-Based Signature Scheme (DABS). According to [40], we construct the DABS scheme that consists of five algorithms: Setup, Keygen, Sign, Verify, and Trace. Select a random security parameter λ; our scheme works as follows:

(i) *Setup* (λ). The algorithm takes a random secure parameter λ as input, and it outputs a master key *MSK*, public key *PK*, and trace key *TK*, where *TK* is used to trace the user identity. Assume that the *PK*

contains the universe of attributes set U, and the default attributes set W.

(ii) *Keygen* $(\mathbf{U_{ID_k}}, \mathbf{MSK}, \mathbf{PK})$. In this algorithm, all attribute authorities AA_i share a pseudorandom function $PRF(\cdot)$. It takes the user's attribute set $U_{ID_k} \subseteq U$, *MSK*, and *PK* as input; each AA computes the attribute private key SK as output.

(iii) $\mathbf{Sign}(\gamma_{(\mathbf{S},\rho)\cup\mathbf{W}}(\cdot), \mathbf{M}, \mathbf{PK}, \mathbf{SK}, \mathbf{ID_k})$. The algorithm takes input signing predicate $\gamma_{(S,\rho)\cup W}$ and shared message *M*, PK, SK, and ID_k, where (S, ρ) is generated according to the access control strategy of the data owner, *S* is a share generating matrix, ρ is a map function as shown in Definition 3, and *W* is a default attribute set. The algorithm outputs a signature δ.

(iv) *Verify* $(\delta, \mathbf{PK})$. The algorithm takes δ and PK as input and outputs a Boolean value.

(v) *Trace* $(\delta, \mathbf{PK})$. The algorithm takes input the signature δ and trace key *TK* and outputs the signer identity ID_k.

Batch Processing. (Definition 4 [41]): giving $\text{BSetup}(\lambda) \longrightarrow (q, g_1, g_2, G_1, G_2, G_T, e)$, where *q* is a prime, λ is a security parameter, and $n = \text{poly}(\lambda)$ $\forall j \in [1, n], A \in \{G_{i|i=1,2,T}\}$ and $u = (u_1, u_2, \ldots, u_n) \in Z_q^*$, if $\prod_{j=1}^{n} A^{u_j} = \prod_{j=1}^{n} Y^{(j)u_j}$, then $A = Y^{(j)}$.

2.5. Security Definitions. This part introduces the security definitions. The ABS scheme supports characters of anonymity and noncollusion. In terms of anonymity, it usually includes computational anonymity and unconditional full anonymity. [16] supports computational anonymity, where the adversary can access the user identity with unlimited computing power. While to the characteristic of unconditional full anonymity, giving a sufficient signature with an access control strategy; adversary Å has unlimited computing power and accesses any users' attribute key. Still, there is no poly(*t*)-time algorithm Λ to reveal the signer's attributes information from the signing predicate. Noncollusion means no poly(*t*)-time for adversaries to forge the legitimate signature with a set of complementary attributes.

2.5.1. Unconditional Full Anonymity. Our scheme supports *unconditional full anonymity* if no adversary Å can win the following games with non-negligible advantagesε.

(i) *Setup*. An adversary Å selects a random signing predicate $\gamma_{(S,\rho)\cup W}(\cdot)$. The simulator $\mathbb{C}$ calls the algorithm Setup(.) and returns the public key *PK* and master key *MSK* to Å. Å can construct a key for any AA_i.

(ii) *Challenge*. In this phase, the Å chooses a random message M' and two attribute sets $\{U_{ID_i}\}_{i=1,2}$, where $\{U_{ID_i}\}_{i=1,2}$ satisfies the signature predicate $\gamma_{(S,\rho)}'$. The Å sends two tuples (M', U_{ID_1}) and (M', U_{ID_2}) to $\mathbb{C}$. The $\mathbb{C}$ calls the algorithm Keygen(.) and returns private keys SK_{ID_1} and SK_{ID_2}. Then $\mathbb{C}$ chooses a bit, $b \in \{0, 1\}$, signs the message (M', U_{ID_b}) as signature δ^{ID_b}, and sends it to the Å.

(iii) *Guess*. The Å outputs the guess result b' of b and wins the game only if $b' = b$.

2.5.2. Noncollusion. Our scheme can defend against collusive attack under adaptive selective message and predicate attacks if there is no adversary Å (capable of unlimited computing power) can win the following games with the non-negligible advantages ε.

(i) Initial.

(ii) Assume $\gamma_{(S',\rho)\cup W}(\cdot)$ is a mini-subset of $\gamma_{(S,\rho)\cup W}(\cdot)$, and $\gamma_{(S,\rho)\cup W}(S', \rho) = 1$. Let the compromised AA group as $\text{SA} = \{AA_1, AA_2, \ldots, AA_{t-1}\}$. For the predicate $\gamma_{(S',\rho)\cup W}(\cdot)$, the adversary Å can forge the signature only if corrupted by another AA_t. So, the attack effect of the DABS scheme can be reducible to attack the AA_t. Assume that the simulator $\mathbb{C}$ chooses a default attributes set W_t from AA_t.

(iii) Setup.

(iv) $\mathbb{C}$ calls Setup(.) and sends *PK*to adversary Å.

(v) Query.

(vi) Å queries the random oracle H, H_2, *SK* and signature from $\mathbb{C}$.

(vii) Challenge.

(viii) The adversary Å challenges the security under a collusive attack. It chooses two random users (ID_0, ID_1) with the attribute set U_{ID_0} and U_{ID_1}, respectively, where $\gamma_{(S',\rho)\cup W}(U_{ID_0} \cup W) \neq 1$, $\gamma_{(S',\rho)\cup W}(U_{ID_1} \cup W) \neq 1$ and $\gamma_{(S',\rho)\cup W}(U_{ID_0} \cup U_{ID_1} \cup W) = 1$.

(ix) The adversary Å requires to query private key of $(U_{ID_0} \cap U_t, ID_0)$ and $(U_{ID_1} \cap U_t, ID_0)$; $\mathbb{C}$ returns the $SK_{ID_0,t}$, $SK_{ID_1,t}$to Å, respectively. So, the combined-key is $SK_{ID,t}' = SK_{ID_0,t} \cup SK_{ID_1,t}$.

(x) Forgery.

Å constructs a signature $\{\gamma_{(S',\rho)\cup W}(U_{ID_0} \cup U_{ID_1} \cup W'), M', \delta'\}$ and will win the game only if

(1) Verify $(\gamma_{(S',\rho)\cup W}(U_{ID_0} \cup U_{ID_1} \cup W'), M', \delta') =$ true.

(2) Å cannot query any private key of the attribute set s' where $\gamma_{(S_s',\rho)\cup W}(s') = 1$.

(3) Å cannot query any private key of the attribute set s' where $\gamma_{(S',\rho)\cup W}(s') = 1$.

3. Cross-Department Secures Data Sharing in the Food Industry via Blockchain-Cloud Fusion Scheme

3.1. System Model. Assume the enterprises, regulators, and neutral institutions (such as food commonweal organization) in the whole food supply-chain hope to share data to promote supervision of food quality and public safety incidents. They provide a cloud server as an Attribute Authority server (AA)

to store the transmitted data and verify them. Regulators and enterprise legal persons are registered with the regulator AA server, and employees register with the enterprise AA server. Assume neutral institutions' servers are semicredible that would not eavesdrop or backup users' registration information. The common consumers, including food enterprise employees, can enter the system by registering with any neutral institution, release supervision information anonymously based on the DABS algorithm (detailed in Section 3.3), and realize social co-governance of all links in the food supply chain. Food enterprises own the AAs, regulators, and neutral institutions independently. To standardize terms, we use data owners as the data providers in the following, including enterprises, regulators, and neutral institutions. As Figure 2 shown that our system consists of four modules: cloud network, DABS, blockchain network, and application group. The system details are as follows:

(i) *Cloud Network Module*. It consists of AA servers' group to support user identity verification, user management, and data storage. Each AA server is selected from the existing system architecture in the food supply chain. AA server is operated and maintained independently by data owners, which also is used as shared data services to support food quality supervision and traceability. Likely, farms can share data in the cultivation server; storage enterprise shares data with the warehouse server, etc. Processing enterprise shares data with the process server; regulators share sampling inspection reports and monitoring data with the regulator server. It uploads the shared data signature and index from food enterprises and regulators to the blockchain network. And, it shares consumers' report data with DABS signature to the blockchain.

(ii) *DABS Module*. It has the security characteristics of unconditional full anonymity and noncollusion (as shown in Section 4). It supports user authentication, secret key assignment, digital signature, and traceability. The AA server generates an anonymous private key for the user with the fuzzy attribute set if the registration information is authenticated. Data owners share data with access control strategy and private key (sign algorithm is detailed in Section 3.3). Furthermore, it provides a way for regulators to trace down the person responsible for rumors and incidents in the food industry (trace algorithm, as shown in Section 3.3), which is helpful to purify the system network. The DABS module is deployed in the AA servers, which works together to maintain the system's stability and security.

(iii) *Blockchain Network Module*. It receives the data from data owners' server, consensus validation with PBFT [39] that more than 2/3 of the servers acknowledge the validity of data, and store the data block into the blockchain (the block structure as shown in Figure 2). Besides, it supports co-governance and traceability of food safety via smart contract (detailed in Section 3.4). If the report is useful, it will give a reward in return, conducive to motivating consumers to participate actively. With the robustness feature of PBFT, the blockchain network can resist no more than 1/3 of the server's failure attack. This module can be deployed in AA servers to save the system development cost.

(iv) *Application Group Module*. It composes of consumers and regulators. Any system user should register firstly by AA server. Consumers can query any quality information of food supply-chain as needed from the blockchain network. The system will then get and return the data from the data owner's server by the data index in the data block. Besides, consumers can take part in food safety supervision. Regulators have the power of supervision to monitor the whole food supply-chain and hold responsible people, including timely warning and accountability of food safety incidents.

3.2. Threat Model and Design Goal. The adversaries can eavesdrop on the public channel's information, including signature and signing predicate. Besides, there are dishonest server groups that are allowed to collude to infer the signature's user identity.

Based on the above threat model, the food supply-chain regulation system hopes to achieve the following goals.

(1) *Privacy Protection*. System user privacy information can be deduced by statistical analysis [13]. So, the system should have the characteristic of *unconditional full anonymity*. It can resist the adversary's statistical analysis and is secure when no more than N $1/N$ of the collusion servers.

(2) *System Availability*. In this work, the system availability includes two aspects. (a) System robustness: on the one hand, it works only if no more than 1/3 servers in the blockchain network fail; on the other hand, it is Strong (*N-1*) for AA servers that it works as long as more than one server is honest. (b) Traceability: it provides a solution to track the person responsible for rumors and food safety incidents.

3.3. Proposed DABS Scheme

Setup (λ). Let G and G_T be two cyclic multiplication groups of composite order $N = pq$ and the bilinear map e: $G * G \longrightarrow G_T$, where p and q are two large prime numbers. The construction also enables to work on asymmetric pairing groups, where e: $G_1 * G_2 \longrightarrow G_T$, and $G_1 \neq G_2$. Denote G_q is the subgroup of order q in G. The universal set of attributes $\boldsymbol{U} = \{U_1, U_2, U_3, \ldots, U_n\}$ are managed by the distribution Attribute Authorities $Group\{\mathrm{AA}_i|_{i=1,2,\ldots,n}\}$. Each AA_i monitors an attribute subset $U_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,|U_i|}\}$ and issues the corresponding attribute private key to users. Define the default attributes set $\mathbf{W} = \{W_1, W_2, W_3, \ldots, W_n\}$, where $W_i = \{j_{i,1}, j_{i,2}, \ldots, j_{i,|W_i|}\}$. Select two collision-

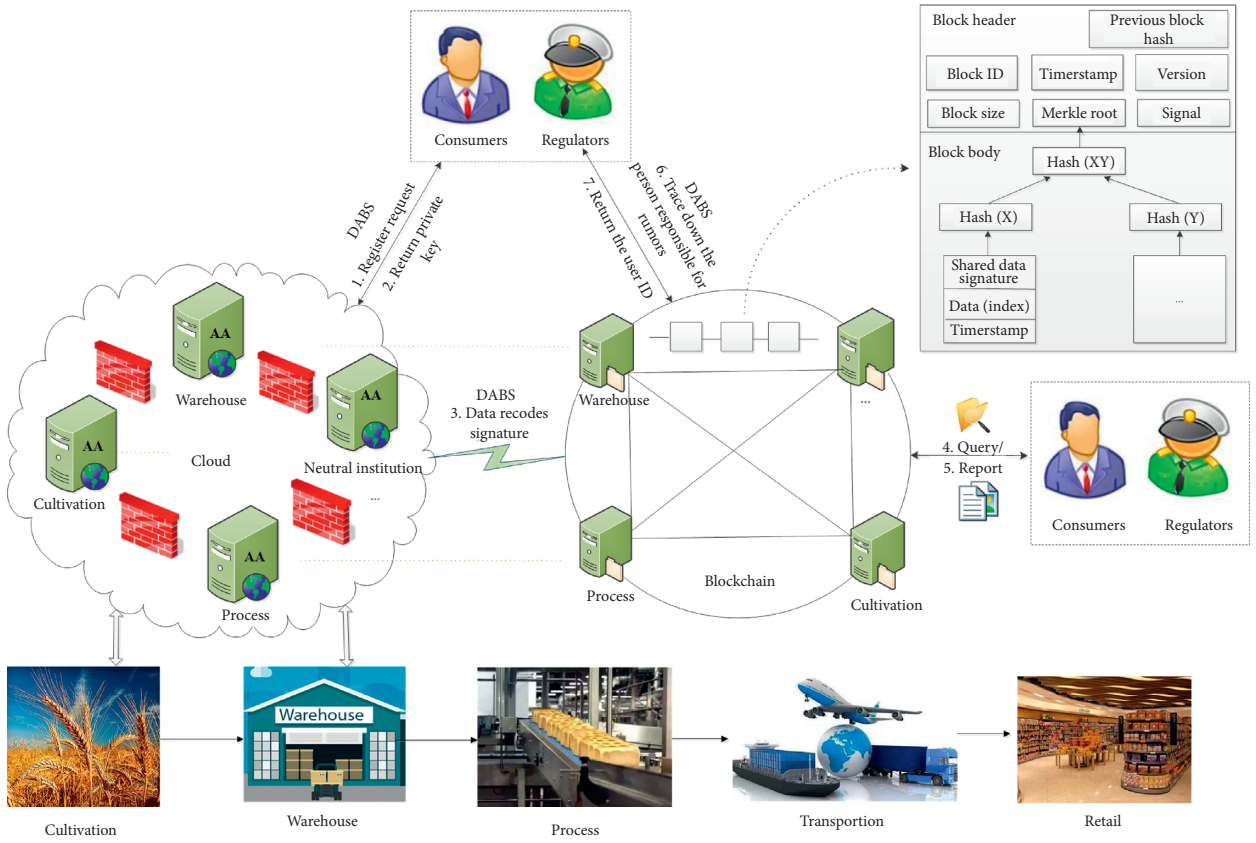

FIGURE 2: The food supply-chain regulation system model based on blockchain-cloud fusion scheme. It remains the existing system architecture and industry metadata owned by data owners. Besides, it enables consumers to know the food quality information throughout the whole supply chain. It provides a safe way for any consumer to social co-governance of food safety without fear of cyber-violence. Furthermore, the regulators can track rumor-mongers.

resistant cryptographic hash function $H, H_2: 0,1^* \longrightarrow G$. Select random generator $g \in G$, exponent $\tau \in Z_P^*$, and compute $T = g^\tau$. Select random exponent $x_i \in Z_P^*$ for each AA_i and compute $P_i = g^{x_i}$. Select random parameter $t_{i,j} \in Z_P^*$ for each attribute. Select generator $u', u_1, u_2, \ldots, u_k \in G$ and $h \in G_q$. So, it generates the public key PK, the master key MSK for AA $Group$, and the trace key TK as follows:

$$\begin{aligned} \mathrm{PK} &= \langle e, g, h, T, \{P_i\}_{i=1,\ldots,n}, \mathbf{W}, H, H_2, u', u_1, u_2, \ldots, u_k \rangle, \\ \mathrm{MSK} &= \langle \{x_i\}_{i=1,\ldots,n}, \{t_{i,j}\}_{i=1,\ldots,n, j \in U_j} \rangle, \\ \mathrm{TK} &= q. \end{aligned} \tag{1}$$

Keygen $(\mathbf{U_{ID_k}}, \mathbf{MSK}, \mathbf{PK})$. In this algorithm, all attribute authorities AA_i share a pseudorandom function $\mathrm{PRF}(\cdot)$. Assume the user ID possesses an attribute set U_{ID}. The AA_i calculates $\Gamma_{ID_{i|i=1,\ldots,n}} = \mathrm{PRF}(ID)$, $D_{0,i|i \in \mathrm{Group}} = g^{\Gamma_{ID_i}}$, and $D_{1,i|i \in \mathrm{Group}} = g^{-x_i}$. For attribute $j \in (U_{ID} \cup W_i)$, it computes $D_{2,i,j|i \in \mathrm{Group}} = H(j)^{t_{i,j}} D_{0,i}$. So, the anonymous private key of the user ID_k is

$$\mathrm{SK} = \langle \{D_{0,i}, D_{1,i}, D_{2,i,j}\}_{i=1,\ldots,n, j \in (U_{ID} \cup W_i)} \rangle. \tag{2}$$

Sign $(\gamma_{(\mathbf{S},\rho) \cup \mathbf{W}}(\cdot), \mathbf{M}, \mathbf{PK}, \mathbf{SK}, \mathbf{ID})$. In this algorithm, the signer ID_k sets an access control strategy $U_{ID_k}' \neq \mathrm{NUll}$ to message M. The access control strategy is $\gamma_{(S,\rho) \cup W}(\cdot) = U_{ID}' \cup W$. The algorithm constructs an LSSS access matrix S_{l*m} with an injective function ρ that maps each row of the matrix S to an attribute of U_{ID}'.

(i) The algorithm randomly chooses a parameter $\varepsilon_i \in Z_P^*$ for every bit of $ID = \{\mathrm{u}[1], \mathrm{u}[2], \ldots, \mathrm{u}[\mathrm{k}]\} \in \{0,1\}^k$, and $c_i = u_i^{u[i]} h^{\varepsilon_i}, \pi_i = (u_i^{2u[i]-1} h^{\varepsilon_i})^{\varepsilon_i}$. Define $\varepsilon = \sum_{i=1,\ldots,k} \varepsilon_i$ and

$$C = u' \prod_{i=1,\ldots,k} c_i = u' \prod_{i=1}^{k} \left(u_i^{u[i]} h^{\varepsilon_i}\right) = u' \prod_{i=1}^{k} u_i^{u[i]} h^{\varepsilon}. \tag{3}$$

The c_i and π_i can prove that Cis well-formed.

(ii) Choose a random parameter $s \in Z_P^*$ and a random vector $v = \{s, v_1, v_2, \ldots, v_{m-1}\} \in \{Z_P^*\}^m$. Define $\lambda_i = S_i \cdot v$, where $S_i$is the ith row of S. Choose a constant $\omega_i \in Z_P^*$ and $\sum_{x \in \rho_{(S_i)}} (\omega_i * S_i) = 1$.

(iii) $\forall$ attribute $x \in \gamma_{(S,\rho) \cup W}$ and select random parameter $r_{i,j} \in Z_P^*$.If $x \in \rho_{(S_i)}$, calculate $\delta_{1,x} = g^{r_{i,j}} D_{2,i,j} T^{\lambda_i}$; If $x \in W$, calculate $\delta_{1,x} = g^{r_{i,j}}$.

(iv) Select fuzzy factors $\forall \gamma \in Z_P^*$ for userID_k and calculate

$$\delta_2 = H_2(M)^{\gamma} \prod_{x\in\rho_{(S_i)}} (\delta_1)^{\omega_i} \prod_{i\in\text{Group}} \prod_{x\in W} \left(P_i D_{1,i} \delta_1\right). \quad (4)$$

(v) Calculate

$$\delta_3 = e\left(g, H_2(M)^{-\gamma} \prod_{x\in\rho_{(S_i)}} \left(g^{r_{i,j}} D_{2,i,j}\right)^{-\omega_i}\right), \quad (5)$$

$$\delta_4 = e(g,T)^{s}.$$

(vi) So, the signature is

$$\delta = \langle \{\delta_1\}_{x\in W}, \delta_2, \delta_3, \delta_4, \{c_i\}_{i=1,\ldots,k}, \{\pi_i\}_{i=1,\ldots,k} \rangle. \quad (6)$$

Verify $(\delta, \mathbf{PK})$. The algorithm takes the signature δ and PK as input and outputs the result. According to the equation result $(e(g,\delta_2) * \delta_3)/(\prod_{i\in\text{Group}}(\prod_{x\in W} e(g, \delta_1))) = \delta_4$, if the equation is correct, this scheme accepts the signature δ and output true, or reject δ and output $\perp$.

Trace $(\delta, \mathbf{TK}, \mathbf{PK})$**.** The algorithm takes input the signature δ, trace key TK, and PK and then outputs the signer identity ID. The algorithm describes as follows:

(i) Call algorithm verifies (δ, PK) and checks the signature δ is true or not.

(ii) If δ is true, $\forall c_i$it will check

$$e\left(c_i, c_i g^{-1}\right) = e\left(h^q, \pi_i^{q^{-1}}\right),$$
$$e\left(h^q, g\right) = e\left(h, g^q\right). \quad (7)$$

(iii) Return u[i] = 1 only if the both check pass, else return u[i] = 0. So, the algorithm will output the signer's identity ID.

The effectiveness of algorithm Trace (.) has been proved in detail in [42], and the security of algorithm Trace (.) has been proved in [43], so we would not detail and analysis the algorithm in the paper.

Correctness. This scheme outputs the signature $\delta = \langle \{\delta_1\}_{x\in W}, \delta_2, \delta_3, \delta_4, \{c_i\}_{i=1,\ldots,k}, \{\pi_i\}_{i=1,\ldots,k} \rangle$ for the message M. We can prove the correctness of the scheme as follows:

$$\begin{aligned}
&\frac{e(g,\delta_2)\delta_3}{\left(\prod_{i\in\text{Group}}\prod_{x\in W} e(g,\delta_1)\right)} \\
&= \frac{e\left(g, \prod_{x\in\rho_{(S_i)}} (\delta_1)^{\omega_i} \prod_{i\in\text{Group}}\prod_{x\in W}\left(P_i D_{1,i}\delta_1\right)\right) e\left(g, \prod_{x\in\rho_{(S_i)}}\left(g^{r_{i,j}} D_{2,i,j}\right)^{-\omega_i}\right)}{\prod_{i\in\text{Group}}\prod_{x\in W} e\left(g, g^{r_{i,j}}\right)} \\
&= e\left(g, \prod_{x\in\rho_{(S_i)}} \left(T^{\lambda_i}\right)^{\omega_i}\right) \\
&= e\left(g, g^{\tau s}\right) = \delta_4, \quad \left(\text{According} \sum_{x\in\rho_{(S_i)}} (\omega_i * \lambda_i) = s\right).
\end{aligned} \quad (8)$$

Batch-Verification Processing. According to the scheme [13], we propose a batch-verification processing algorithm to improve the effectiveness of $nO(\cdot)$ to $O(\cdot) + n$. It takes inputs the public key PK and a large number of signatures $\left\{\delta^{ID_1}\delta^{ID_2}, \ldots, \delta^{ID_n}\right\}$, and works as follows:

$$\begin{aligned}
&\frac{e\left(g, \delta_2^{ID_1}\delta_2^{ID_2}, \ldots, \delta_2^{ID_n}\right)\left(\delta_3^{ID_1}\delta_3^{ID_2}, \ldots, \delta_3^{ID_n}\right)}{\prod_{i\in\text{Group}}\prod_{x\in W} e\left(g, \delta_1^{ID_1}, \ldots, \delta_1^{ID_n}\right)} \\
&= e(g,T)^{\sum_{i=1,\ldots,n}^{s}} = \prod_{i=1,\ldots,n} \delta_4^{ID_i}.
\end{aligned} \quad (9)$$

3.4. Social Co-Governance of Food Safety Based on Smart Contract. The smart contract enables to automatic execution of the agreement between the parties without an intermediary. It is helpful to improve the efficiency of information processing and social co-governance of food safety.

(1) *Data Intelligence Verification*. When enterprise servers upload signature $\delta = \langle \{\delta_1\}_{x\in W}, \delta_2, \delta_3, \delta_4, \{c_i\}_{i=1,\ldots,k}, \{\pi_i\}_{i=1,\ldots,k} \rangle$ to the blockchain network, it would autorun the smart contract deployed in blockchain servers and intelligent process verification of the signature, as shown in Algorithm 1.

(2) *Social Co-Governance of Food Safety*. If the quality problem happens in the food supply-chain, any consumer can report it anonymously by the blockchain network. On one hand, smart contract helps regulators timely deal with the potential risk of food safety incidents and investigates the legal liability of the enterprises involved; on the other hand, it will warn someone who spreads rumors. The smart contracts are constructed as shown in Algorithm 2 and 3.

Input: signature δ and public key PK
Output: success or alarm
(1) **Function**: verify (δ, PK)
(2) **If** $e((g, \delta_2) * \delta_3)/(\prod_{i \in Group}(\prod_{x \in W} e(g, \delta_1))) = \delta_4$ **Then**
(3) Writeblockchain (δ)
(4) Authorize (*nextstep*)
(5) **Return** *Success*
(6) **Else**
(7) Writeblockchain (*false*)
(8) **Return** Alarm (*"the signature is invalid"*)
(9) **End If**
(10) **End Function**

Algorithm 1: Automated validation data by enterprises and regulatory agencies.

Input: Food safety *report*, trace key TK and public key PK
Output: Alarm
(1) **Function** Co-governance (*report*, TK, PK)
(2) result = Consensus (*report*)
(3) $\delta \Longleftarrow$ reporter. getDataSign ()
(4) **If** result = true **Then**
(5) $ID \Longleftarrow$ Trace (δ, PK, TK)
(6) Writeblockchain (*report*)
(7) **Return** Alarm (*"the enterprise"* + ID + *"has food safety problems"*)
(8) **Else**
(9) $ID \Longleftarrow$ Trace (δ, PK, TK)
(10) **Return** Alarm (*"the reporter"* + ID + *"posts a malicious information"*)
(11) **End If**
(12) **End Function**

Algorithm 2: Report and accountability for food safety problems.

Input: Food safety report, trace key TK and public key PK
Output: Identity information ID
(1) **Function** Trace (δ, PK, TK)
(2) $ID \Longleftarrow$ null
(3) **If** (Verify (δ, PK) = Success) **Then**
(4) $c \Longleftarrow$ reporter. getListC ()
(5) $\pi \Longleftarrow$ reporter. getListπ ()
(6) $h \Longleftarrow$ PK.getH ()
(7) $i \Longleftarrow 0$
(8) **While** ($i < c$.length)
(9) **If** $(e(c_i, c_i g^{-1})) = e(h^q, \pi_i^{q^{-1}})$ and $e(h^q, g) = e(h, g^q)$ Then
(10) $ID \Longleftarrow ID +$ "1"
(11) **Else**
(12) $ID \Longleftarrow ID +$ "0"
(13) **End If**
(14) **End While**
(15) **End If**
(16) **Return** ID
(17) **END Function**

Algorithm 3: Tracking user identity ID.

3.5. Food Supply-Chain Regulation System Based on Blockchain-Cloud Fusion Scheme. With the blockchain-cloud fusion scheme, the food supply-chain regulation system overcomes the pain points of data sharing in the food industry. It is conductive to optimize the processes of information collection, quality inspection and supervision, and supply and marketing management. This part mainly introduces the system workflow.

We will describe the system workflow from two aspects of data sharing and data consumption. The shared data consist of consumer reports and food industry shared data.

3.5.1. Industry Data Sharing. On receiving the shared data from enterprises and regulators in the food supply-chain, the system workflow is as shown in Figure 3.

We will take the food enterprise data sharing as an example to detail the system workflow as follows:

(1) Any system user should authenticate and register by the AA server first; then, the AA server would verify its identity and generate an anonymous private key. Each workflow needs this step so that we would not detail it again in other workflows.

(2) After authentication is completed, the enterprise manager can share data by defining a signing predicate on demand and then sending it to the enterprise AA server.

(3) The AA server generates an anonymity signature and sends the signature and the index of shared data to the blockchain network.

(4) Blockchain server firstly determines the validity of data via smart contract and generates a data block. Validate and then broadcast the block to make a consensus decision with PBFT algorithm.

(5) If more than 2/3 of the servers agree, the new data block will be stored in the blockchain.

3.5.2. Consumer Report. To promote social co-governance in food safety, the system provides an anonymous supervision report function for consumers and gives a reward in return. The system workflow of consumer reports is similar to the front workflow—the workflow is shown in Figure 4.

(1) Consumers in the system report the food quality problems anonymously. They can define a signing predicate to generate an anonymous signature.

(2) The AA server would share the report content and signature to the blockchain. Then, verify and generate a new data block with a signal tag. Make a consensus decision and store the block into the block chain.

(3) The blockchain network will send an alert message to regulators to make a decision. The system will give a reward to the consumer in return if the alarm is validated for regulators.

In addition, data consumption consists of consumers inquiring about food quality on-demand and accountability.

3.5.3. Inquire Food Quality on Demand. The system consumers inquire about food information, as shown in Figure 5.

(1) Consumers can inquire about the quality information (including quality inspection report, source, process, and transport) of each link's raw material in the food supply-chain from a blockchain network.

(2) It analyzes the data source index from the data block and gets the target data from the data owner's server and then shows it to the consumer as a basis for food quality evaluation.

3.5.4. Accountability. Due to blockchain's tamper-proofing feature, the system provides traceability for regulators to track the person responsible for rumors and food quality incidents. We take tracking a rumor-monger as an example to detail the system workflow as follows (as shown in Figure 6).

(1) The regulator chooses a rumor to track the monger from the blockchain network. The system will authenticate the regulator and record the action.

(2) Then, the system analyzes the rumor record, generates the rumor-monger's ID, and returns it to the regulator, which can serve as a basis for the regulator to law enforcement [44].

4. Security Analysis

Blockchain-cloud fusion scheme inherits some essential characteristics of blockchain and cloud service to protect the system's data. The tamper-proof feature of blockchain ensures data reliability. PBFT-based consensus mechanism can improve the system robustness, and DABS algorithms protect the system's safety and stability. In this work, system security mainly prevents the leakage of user privacy. Since the data interaction in the scheme is based on the DABS algorithm, and user privacy information is processed and generated by DABS, the DABS algorithm's security is the most critical factor for the protection of the scheme. We will mainly analyze the security of the DABS algorithm in this section.

4.1. Theorem 1: Unconditional Full Anonymity

Proof. This scheme can construct a sufficient signature if the signature attributes satisfy the signature predicate $\gamma_{(S,\rho)\cup W}(\cdot)$. With the predicate subset and default attributes mixed in the signature, the adversary Å cannot get signer attributes from the signature predicate. So, our scheme supports unconditional full anonymity if the adversary Å cannot get user identification information. According to the schemes [45], we construct the simulation as follows.

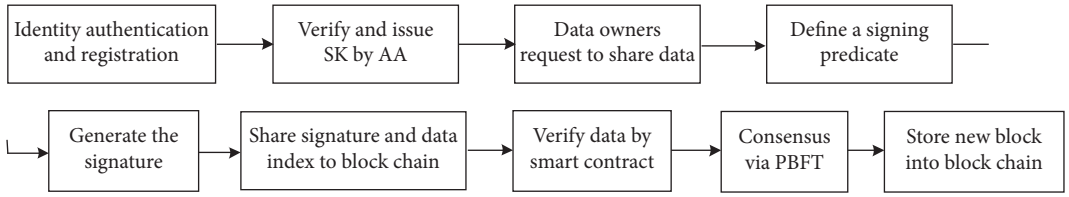

Figure 3: System workflow of industry data sharing.

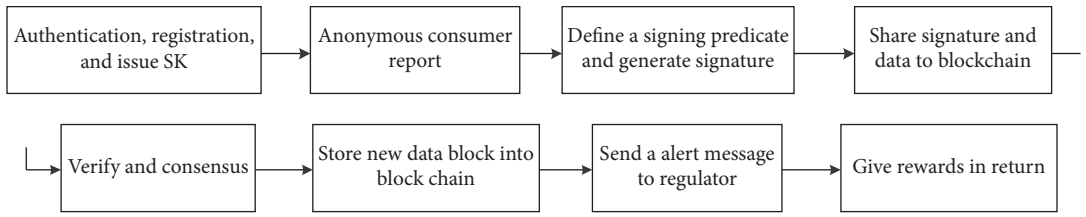

Figure 4: System workflow of consumer report.

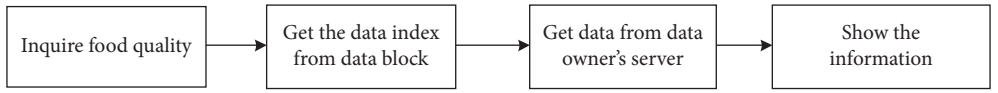

Figure 5: System workflow of inquire food information.

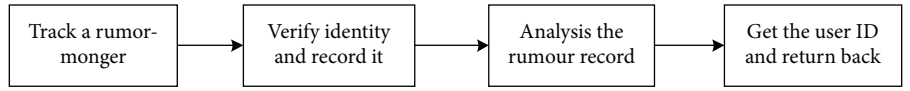

Figure 6: System workflow of accountability.

(1) Setup:

An adversary Å challenges the access control predicate $\gamma_{(S,\rho)\cup W}(\cdot)$. The simulator $\mathbb{C}$ calls algorithm Setup() and outputs the *PK*, *MSK*, and *TK*. Then, it public the *PK* and *MSK* to Å. The adversary Å can construct any private key.

(2) Challenge:

In this phase, the Å chooses a random message M' and two attribute sets $\{U_{ID_i}\}_{i=1,2}$, where $\{U_{ID_i}\}_{i=1,2}$ satisfies the signature predicate $\gamma_{(S,\rho)}{}'$ and then the Å transmit the (M', U_{ID_1}) and (M', U_{ID_2}) to $\mathbb{C}$. The $\mathbb{C}$ works as follows:

(i) The $\mathbb{C}$ calls the algorithm Keygen() and returns the private key: $SK^{ID_1} = \{D_{0,i}^{ID_1}, D_{1,i}^{ID_1}, D_{2,i,j}^{ID_1}\}$ and $SK^{ID_2} = \{D_{0,i}^{ID_2}, D_{1,i}^{ID_2}, D_{2,i,j}^{ID_2}\}$.

(ii) The simulator chooses a bit, $b \in \{0, 1\}$, signs the message with (M', U_{ID_b}) (detailed as Section 3.3), and outputs the signature δ^{ID_b} to Å:

$$\delta^{ID_b} = \{\delta_1^{ID_b}, \delta_2^{ID_b}, \delta_3^{ID_b}, \delta_4^{ID_b}, \{c_i^*\}_{i=1,\ldots,k}, \{\pi_i^*\}_{i=1,\ldots,k}\}. \tag{10}$$

(iii) The Å guesses the b' of b in $\delta^{ID_b} = \{\delta_1^{ID_b}, \delta_2^{ID_b}, \delta_3^{ID_b}, \delta_4^{ID_b}\}$, where $\delta_1^{ID_b} = \{g^{r_{i,j}}\}_{x\in W}$

$$\delta_2^{ID_b} = H_2(M')^{\gamma} \prod_{x\in\rho_{(S_i)}} \left(\delta_1^{ID_b}\right)^{\omega_i} \prod_{i\in Group} \left(\prod_{x\in W} P_i D_{1,i} \delta_1\right)$$

$$\delta_3^{ID_b} = e\left(g, H_2(M)^{-\gamma} \prod_{x\in\rho_{(S_i)}} \left(g^{r_{i,j}} D_{2,i,jID_b}\right)^{-\omega_i}\right)$$

$$\delta_4^{ID_b} = e(g, T)^{s_{ID_b}}. \tag{11}$$

(3) Guess:

The adversary submits a guess b' of b. If $b' = b$, Å wins the game, which means the scheme cannot support the *unconditional full anonymity* security. Next, we discuss why the Å cannot win the game.

Assume the simulator $\mathbb{C}$ selects $b = 1$, signature predicate $\gamma_{(S,\rho)_{ID_1}\cup W_{ID_1}}(\cdot)$, and random α_1 and s_1.

Then, it signs a message with (M', SK_{ID_1}) and outputs $\delta^{ID_1} = \{\delta_1^{ID_1}, \delta_2^{ID_1}, \delta_3^{ID_1}, \delta_4^{ID_1}\}$ as follows:

$$
\begin{aligned}
\delta_1^{ID_1} &= \left\{g^{r_{i,j}}\right\}_{x\in W} \\
\delta_2^{ID_1} &= H_2\left(M'\right)^{\gamma_1} \prod_{x\in\rho_{(S_i)}} \left(g^{r_{i,j}} g^{\tau\lambda_i} H(x)^{t_x} g_{ID_1}^{\Gamma_{ID_{i,k}}}\right)^{\omega_i} \prod_{i\in\text{Group}} \left(\prod_{x\in W_{ID_1}} g^{r_{i,j}}\right) \\
&= g^{\tau s_1 + \sum_{i\in\text{Group}} \Gamma_{ID_{i,1}} + \sum_{i\in\text{Group}, x\in\rho_{(S_i)}\cup W_{ID_1}} r_{i,j}} H_2\left(M'\right)^{\gamma_1} H(x)^{\sum_{x\in\rho_{(S_i)}} t_x} \\
\delta_3^{ID_1} &= e\left(g, H_2\left(M'\right)^{-\gamma_1} g^{\sum_{i\in\text{Group}} \Gamma_{ID_{i,1}}} H(x)^{\sum_{x\in\rho_{(S_i)}} t_x}\right) \\
\delta_4^{ID_1} &= e(g,T)^{s_{ID_b}} = e(g,g)^{\tau s_1}.
\end{aligned} \tag{12}
$$

As δ^{ID_1} shows that only $\delta_2^{ID_1}$ and $\delta_3^{ID_1}$ involve the user identification information, if there exists $\gamma_{(S,\rho)_{ID_1}\cup W_{ID_1}}(U_{ID_1}) = \gamma_{(S,\rho)_{ID_2}\cup W_{ID_2}}(U_{ID_2})$, $\{t_x^1\}_{x\in\rho^1_{(S_i)}} = \{t_x^2\}_{x\in\rho^2_{(S_i)}}$, $\{\gamma^1_{ID_1}\} = \{\gamma^1_{ID_2}\}$, and $\sum_{i\in\text{Group}}\Gamma_{ID_{i,2}} + \sum_{x\in\rho^2_{(S_i)}} r_{i,j,} \ 2 = \sum_{i\in\text{Group}}\Gamma_{ID_{i,1}} + \sum_{x\in\rho^1_{(S_i)}} r_{i,j,1}$, the simulator $\mathbb{C}$ can generate the same signature $\delta^{ID_1} = \delta^{ID_2}$ whatever a bit b chose. So, our scheme satisfies absolute full anonymity and does achieve perfect privacy. □

4.2. Theorem 2: Noncollusion. Our scheme can provide defense against collusive attacks under adaptive selective messages and predicate attacks.

Proof. We describe our DABS scheme's security model by the next game between simulator $\mathbb{C}$ and adversary Å. The security model allows the adversary to query for any private keys that cannot be used to sign the challenge Message M [18]. Assume the adversaries can corrupt authorities statically, and the key queries are adaptively [46].

Assume there is a poly(t)-time algorithm Λ for the adversary Å can break our scheme with non-negligible advantages ε under the adaptive selection message and collusive attack. Define parameters $q_{\text{PRF}}, q_H, q_{H_2}, q_k$, and q_s used to label the query number of random oracles PRF, H, and H_2, generate the SK and signature, respectively. So, there is a poly(t)-time algorithm Λ that can deal with the CDH problem with a non-negligible advantage $\varepsilon\prime = \varepsilon / \left(q_H q_{H_2} \prod_{i\in SA\cup t}\left(q_i\left(\frac{|W_i|}{|U_i|}\right)\right)\right)$. The security simulation proceeds as follows:

(1) Initial:

Assume predicate $\gamma_{(S',\rho)\cup W}(\cdot)$ is a mini-subset of $\gamma_{(S,\rho)\cup W}(\cdot)$ and $\gamma_{(S,\rho)\cup W}(S',\rho) = 1$, where the attributes in $\gamma_{(S',\rho)\cup W}(\cdot)$ are managed by $\{\text{AA}_i\}_{i=1,\ldots,t}$. Let $\rho(S') = \sum_{i=1}^{t}\rho(S_i')$, and define the attribute set of users ID_k is U_{ID_k}. Assume the corrupted AA group is $\text{SA} = \{\text{AA}_1, \text{AA}_2, \ldots, \text{AA}_{t-1}\}$. The adversary Å can forge the signature only if another AA_t is corrupted. So, the collusive attack effect of our scheme can be reducible to attack a signal node AA_t. Define q_t as the node AA_t attacked probability and $q_t = 1/(n-t+1)$. The simulator $\mathbb{C}$ chooses a default attributes set W_t for AA_t.

(2) Setup:

The simulator $\mathbb{C}$ selects random exponent $\tau\in Z_P^*$. Send $P_t = g^{x_t}$ and $T = g^{\tau}$ to adversary Å Select generators $u', u_1, u_2, \ldots, u_k \in G, h \in G_q$ and publish to Å.

(3) Query:

The adversary Å can query by random oracle H,H_2, SK, and signature. The simulator $\mathbb{C}$ maintains the empty list $L_{\text{PRF}}, L_H, L_{H_2}$, and L_{key}; the processes is as follows:

(i) PRF query: the simulator $\mathbb{C}$ maintains the list L_{PRF} to store the result $(ID_i, \text{AA}_t, \text{PRF}_t(ID_k), \Upsilon_k)$. When receiving a query request (ID_i, AA_t) from Å, $\mathbb{C}$ checks L_{PRF} and returns the result if the request had been received. Otherwise, $\mathbb{C}$ chooses random parameters $a_{k,t}, \varrho_k \in Z_P^*$, set $\text{PRF}_t(ID_k) = a_{k,t}, \Upsilon_k = g^{\varrho_k}$ and publishes to Å and then adds $(ID_i, \text{AA}_t, \text{PRF}_t(ID_k), \Upsilon_k)$ into L_{PRF}.

(ii) H Query: the simulator $\mathbb{C}$ maintains the list L_H to store the result $(j_t, H_{t,j})$. When receiving a query request (j_t) from Å, $\mathbb{C}$ checks L_H and returns the result if the request had been received. Otherwise, $\mathbb{C}$ processes as follows:

(a) If $j_t \in ((U_{ID_k} \cap U_t) \cup W_t)$, it chooses a random parameter $\varphi_{t,j}\in Z_P^*$, set $H(j_t) = g^{\varphi_{t,j}}$, publishes to Å and then adds $(j_t, H_{t,j})$ into L_{PRF}.

(b) If $j_t \notin ((U_{ID_k} \cap U_t) \cup W_t)$, it chooses random parameters $\varphi_{t,j}, \beth_{t,j}\in Z_P^*$, publishes $H(j_t) = g^{\varphi_{t,j}} g^{\beth_{t,j}}$ to Å and then adds $(j_t, H_{t,j})$ into L_{PRF}.

(iii) H_2 query: the simulator $\mathbb{C}$ maintains the list L_{H_2} to store the result $(M_i, H_{2,i})$ and chooses random parameter $\varepsilon \in [1, q_{H_2}]$. When receiving a query request (M_t) from Å, $\mathbb{C}$ checks L_{H_2} and returns the result if the same value has been queried. Otherwise, C processes as follows:

If $t = \varepsilon$, it chooses $\eta_t \in Z_P^*$ and publishes $H_{2,t} = g^{\eta_t}$ to Å. Then, it adds $(M_t, H_{2,t})$ into L_{H_2}.

If $t \neq \varepsilon$, it chooses $\varphi_t', \eta_t \in Z_P^*$ and publishes $H_{2,t} = g^{\eta_t} g^{\varphi_t'}$ to Å. Then, it adds $(M_t, H_{2,t})$ into L_{H_2}.

(iv) *SK* query: the simulator $\mathbb{C}$ maintains the list L_{key} to store the result $\{(U_{ID_k} \cap U_t), ID_k, SK_k, P_t\}$. When receiving a query request $(U_{ID_k} \cap U_t, ID_k)$ from Å, $\mathbb{C}$ checks L_{key} and returns the result if the same value has been queried. Otherwise, $\mathbb{C}$ processes as follows:

(a) If $\gamma_{(S_\rho', \rho) \cup W}(U_{ID_k} \cup W_t) \neq 1$, it returns $SK_{ID_k,t} = \{D_{0,t} = g^{\lambda_{ID_{i,t}}} = g^{a_k,} \quad t, \quad D_{1,t} = g^{-x_t}, D_{2,t,j} = H(j)^{t_{i,j}} D_{0,i} = g^{\varphi_{t,j} t_{i,j}} D_{0,i} | \forall j \in ((U_{ID_k} \cap U_t) \cup W_t)\}$ and $P_t = g^{x_t}$ *to* Å. Then, adds $\{(U_{ID_k} \cap U_t), ID_k, D_{t,j,j \in ((U_{ID_k} \cap U_t) \cup W_t)}, P_t\}$ into L_{key}.

(b) If $\gamma_{(S_\rho', \rho) \cup W}(U_{ID_k} \cup W_t) = 1$, it stops and defines the process event as $\mathbf{E}_1$.

(v) Signature query: on receiving a signature query of (U_{ID_k}, ID_k, M_t) from Å, $\mathbb{C}$ processes it as follows:

(a) If $\gamma_{(S_\rho', \rho) \cup W}(U_{ID_k} \cup W_t) = 1$, it checks $((U_{ID_k} \cap U_i), ID_k, SK_k, P_i)_{i \in \text{Group}}$, $(M_t, H_{2,i})_{i \in \text{Group}}$ and $(j_i, H_{i,j})_{i \in \text{Group}, j \in U_{ID_k} \cup W_t}$, then calls the algorithm Sign(.), and returns δ^{ID_k} to Å.

(b) Else, it stops and defines the process event as $\mathbf{E}_2$.

(7) Challenge:

(vi) The adversary Å challenges the security under a collusive attack. It chooses any two users (ID_0, ID_1) with the attribute sets U_{ID_0} and U_{ID_1}, respectively, where $\gamma_{(S', \rho) \cup W}(U_{ID_0} \cup W) \neq 1$, $\gamma_{(S', \rho) \cup W}(U_{ID_1} \cup W) \neq 1$ and $\gamma_{(S', \rho) \cup W}(U_{ID_0} \cup U_{ID_1} \cup W') = 1$.

(vii) The adversary Å requires to query (ID_0, AA_t) and (ID_1, AA_t), and $\mathbb{C}$ checks L_{PRF} and returns $a_{0,t}, a_{1,t}$ respectively.

(viii) The adversary Å requires to query $(U_{ID_0} \cap U_t, ID_0)$ and $(U_{ID_1} \cap U_t, ID_0)$, and $\mathbb{C}$ checks L_{key} and processes as follows:

(ix) $SK_{ID_0,t} = \{D_{0,t} = g^{a_{0,t}}, \ D_{1,t} = g^{-x_t}, \ D_{2,t,j} = H(j)^{t_{i,j}} g^{a_{0,t}} = g^{\varphi_{t,j} t_{i,j}} D_{0,i} | \forall j \in ((U_{ID_1} \cap U_t) \cup W)\}$ and $P_t = g^{x_t}$.

(x) $SK_{ID_1,t} = \{D_{0,t} = g^{a_{1,t}}, D_{1,t} = g^{-x_t}, D_{2,t,j} = H(j)^{t_{i,j}} D_{1,i} = g^{\varphi_{t,j} t_{i,j}} g^{a_{1,}} \ t | \forall j \in ((U_{ID_1} \cap U_t) \cup W)\}$, $D_{1,t} = g^{-x_t}$, $D_{2,t,j} = H(j)^{t_{i,j}} D_{1,i} = g^{\varphi_{t,j} t_{i,j}} g^{a_{1,t}} | \forall j \in ((U_{ID_1} \cap U_t) \cup W)\}$ and $P_t = g^{x_t}$.

So, the private key of the user ID_1 can be reconstructed as ID_1':

$$\begin{aligned} SK_{ID_1',t} &= \left\{ D_{0,t}' = \left(D_{1,t} \Upsilon_1\right)^{(a_{0,t}/a_{1,t})} = g^{a_{0,t}} g^{\varrho_k'} \right. \\ & D_{1,t}' = D_{1,t} = g^{-x_t}, \\ & D_{2,t,j}' = H(j)^{t_{i,j}} \\ & \left. D_{0,t}' = g^{\varphi_{t,j} t_{i,j}} g^{a_{0,t}} g^{\varrho_k'} | \forall j \in \left(\left(U_{ID_1} \cap U_t\right) \cup W\right) \right\} \\ & P_t = g^{x_t}. \end{aligned} \tag{13}$$

The adversary Å can get the private key is $\{SK_{ID_0,t} \cup SK_{ID_1',t}, P_t\}_{j \in (U_{ID_0} \cup U_{ID_1}' \cup W')}$.

(8) Forgery:

Å constructs a signature $\{\gamma_{(S', \rho) \cup W}(U_{ID_0} \cup U_{ID_1} \cup W'), M', \delta\prime\}$ where $H_{2,t} = g^{\eta_t}$, and $\mathbb{C}$ verifies correctness as follows:

$$e(g, \delta_2') * \frac{\delta_3'}{\left(\prod_{i \in \text{Group}} \prod_{x \in W} e(g, \delta_1')\right)}$$

$$= \frac{e\left(g, \prod_{x \in \rho_{(S_i)_0}} \left(g^{r_{i,j} T^{\lambda_i}} D_{2,i,j}\right)^{\omega_i} \prod_{x \in \rho_{(S_i)_1}} \left(g^{r_{i,j} T^{\lambda_i}} D_{2,i,j}\right)^{\omega_i} \prod_{i \in SA \cup t} \left(\prod_{x \in W} g^{x_t} D_{1,i} g^{r_{i,j}}\right)\right)}{\prod_{i \in SA \cup t} \left(\prod_{x \in W'} e(g, g^{r_{i,j}})\right)} \tag{14}$$

$$* e\left(g, \prod_{x \in \rho_{(S_i)_0}} \left(g^{x_i} D_{1,i}\right)^{-\omega_i} \prod_{x \in \rho_{(S_i)_1}} \left(g^{x_i} D_{1,i}\right)^{-\omega_i}\right)$$

$$= e\left(g, \prod_{x \in \rho_{(S_i)_0}} \left(g^{\tau \lambda_i}\right)^{\omega_i} \prod_{x \in \rho_{(S_i)_1}} \left(g^{\tau \lambda_i}\right)^{\omega_i}\right) = e\left(g, g^{\tau s'}\right) = \delta_4'.$$

Table 2: Notations.

Notations	Description
UA	The universe AA set
$U_{i\|i\in UA}$	The attribute set managed by AA_i
U_{ID_i}	The attribute set of users ID_i
$U_{ID_{i,j}}$	The attribute set of $U_{ID_i} \cap U_j$
$\|p\|$	The length of the element in Z_p
$\|ID_i\|$	The size of the user ID_i
$\|G\|$	The element size in group G
$\|GT\|$	The element size in the group G_T
n	The total number of registered users in the scheme
S_{l*k}	The access control strategy matrix and l is the number of attributes
d	The default attribute set
N	The number of AA server

So, we can get $g^{\tau s'} = \delta_2'\delta_3'/(\prod_{i\in SAUt}\prod_{x\in W_i'}(\delta_{1,x}{}'))$. It means the CDH problem can be solved, and the advantageis $\varepsilon\prime = \varepsilon/\left(q_H q_{H_2}\prod_{i\in SAUt}\left(q_i\binom{|W_i|}{|U_i|}\right)\right)$, where the probability of $H(j_t) = g^{\varphi_{t,j}}$ is $1/q_H, H_{2,t} = g^{\eta_t}$ is $1/q_{H_2}$ and $q_i = 1/(n-i+1)$. □

5. Performance Analysis

To evaluate the performance of the schemes, we present theoretical analysis of storage complexity and experimental simulation of computation efficiency. Assume the group order in our scheme has the same length with the group order of G in comparison schemes. The parameter notation description is shown in Table 2.

5.1. Storage Complexity. The storage complexity is one of the most evaluation indexes of the food supply-chain regulation system. Compared with a traditional cloud-based system, our system's storage cost mainly focuses on the blockchain and DABS algorithm. Since the blockchain network data primarily comes from DABS, we will mainly analyze the DABS algorithm's storage complexity.

As shown in Table 3, we analyze the performance of the DABS scheme by comparing it with GSZ18's scheme [26], LW10's scheme [40], SZW18's scheme [27], YJ13's scheme [47], and RW13's scheme [48]. These schemes adopt LSSS access control strategy except for [GSZ18] scheme with tree access control strategy. The AA storage overhead, which is used to store the master key and AA's secret key, is $(U_i + 1)|p|$ in our scheme. It is significantly less than [LW10], [SZW18], and [GSZ18]. Since the AA in [YJ13] stores all users' private keys to re-encrypt the ciphertext and update information during the revocation, the AA storage overhead will be less in our scheme if the number of users more than half of $|U_i|$. Besides, compared with the normally anonymity for private keys in other schemes, we improve the security of users' private keys to *unconditional full anonymity*.

User storage overhead used to store the users' private key is $(U_{ID_i} + d + 2)|G|$ in our scheme, that is, better than $(2U_{ID_i} + 2)|G| + U_{ID_i}|p|$ in [RW13] and $U_{ID_i}(|G| + |p|) + |G|$ in [GZW18]. The [GZW18] scheme costs communication time $o(N^2)$ to communicate with AA to generate user private key, while it costs $o(N)$ in ours. Further more, our signature size is $2|GT| + (d + 2|ID_i| + 1)|G|$ that is more effective than [RW13] and [GSZ18]. And, our signature storage overhead has nothing with user attributes, which means that the future expansion of the system has little impact on the DABS algorithm. It provides a solution to track the signer identity by storing each parameter of ID_i. If the size of ID_i is set appropriately, the storage complexity of our scheme will be superior to the schemes [YJ13], [SZW18], [RW13], and [GSZ18].

5.2. Computation Efficiency. We implement our scheme, [LW10] scheme, [RW13] scheme, scheme in [49] (named as [BSW07]), and [27] scheme (named as [SUN18]) in window 10 system with an Intel (R) Core(TM) i7-8565U CPU @ 1.8 GHz 1.99 GHz and 8 GB RAM. And, the server is deployed in VMware® Workstation 15 Pro with the configuration shown in Table 4. It uses the Java Pairing-Based Cryptography (PBC) library version 2.0 to implement the access control schemes. We choose an asymmetric elliptic curve where the order p is a 160 bit length prime. Define the size of plaintext, G generator, and GT generator is 128 Byte. We take the average value of 20 experiments as the final experimental result.

It mainly compares the time efficiency of setup, private key generation, signature, and verification. Figure 7(a) describes the comparison of setup time. Our scheme's performance is much better than [LW10] scheme and [SUN18] scheme, because they spend too much time calculating the complex pairing operations. Figure 3(b) describes the comparison of key generation time where the number of default attributes is 5. Our scheme's key generation time is less than [BSW07] scheme and [RW13] scheme. Figure 7(d) describes the comparison of verification time. It shows that our scheme incurs less verification time than others. Besides, our scheme support batch-verification with time complexity $O(\cdot) + n$ compared $nO(\cdot)$ in other schemes. So, if batch signatures are verified, our system's advantage will be more significant.

Figure 7(c) describes the performance comparison of signature algorithms, where the number of default attributes is 5. When the size of user attribute set is less than 25, the advantage of our scheme is not obvious compared with other

Table 3: Comparison among different attribute-based signature schemes.

Scheme	[YJ13]	[LW10]	[SZW18]	[RW13]	[GSZ18]	Ours
AA storage overhead	$(2n+1)\|p\|$	$2U_i\|p\|$	$\sum_{i\in UA}U_i(2\|p\|+\|GT\|+\|G\|)$	$\|p\|$	$(U_i+N+1)\|p\|$	$(U_i+1)\|p\|$
User ID_i storage overhead	$3\|p\|+(\sum_{j\in UA}U_{ID_{i,j}}+2)\|G\|$	$U_{ID_i}\|G\|$	$U_{ID_i}(\|G\|+\|p\|)$	$(2U_{ID_i}+2)\|G\|+U_{ID_i}\|p\|$	$U_{ID_i}(\|G\|+\|p\|)+\|G\|$	$(U_{ID_i}+d+2)\|G\|$
Signature size	$(4l+3)\|G\|$	$(l+1)\|GT\|+2l\|G\|$	$(l+1)\|GT\|+\|p\|$	$S_{l*k}\|p\|+\|\rho\|+\|GT\|+(3l+1)\|G\|$	$\|GT\|+(3U_{ID_i}+l+3)\|G\|$	$2\|GT\|+(d+2\|ID_i\|+1)\|G\|$
Multiauthority	Yes, but it needed a fully trusted CA	No, but scalable	Yes	No	Yes	Yes
Predicate	Monotone	Monotone	Monotone	Monotone	Monotone	Monotone
Authority corruption type	Weak	Strong (N -1)	Strong (N-1)	No	Strong (N-2)	Strong (N-1)

Table 4: Deployment environment.

Elements	Parameters
CPU cores	4
Processor	1
CPU model	Intel (R) core(TM) i7-8565U CPU @ 1.8 GHz 1.99 GHz
RAM	2 GB
SCSI	60 GB

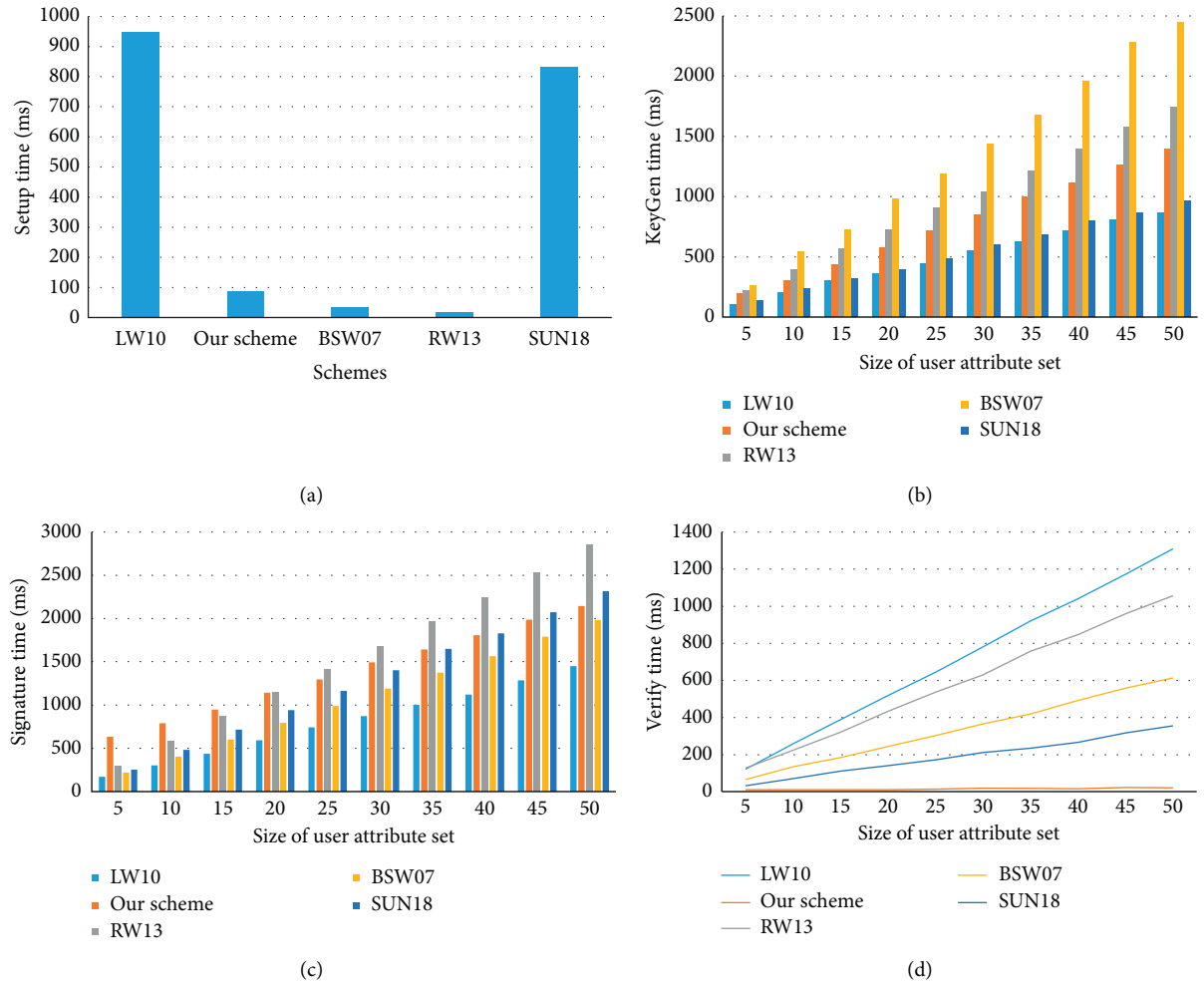

Figure 7: Comparison of computation time. (a) Setup. (b) Key generation. (c) Signature generation. (d) Verification.

schemes. But, once the user attributes size exceeds 25, the advantages of our scheme will gradually emerge. Because, our scheme takes some computational cost in terms of user attributes anonymity and identity tracking, including calculate $2|ID_i|$bilinear operation to establish traceable evidence of user identity, which helps government regulators track down the malicious user. So, our scheme is more suitable for large and complex industry network.

6. Conclusion

We have proposed a food supply-chain regulation system based on a blockchain-cloud fusion scheme. It did not let the source data not out of data owners to protect enterprises' benefit and reserves the original system architecture to reduce the cost. Then, we presented a security DABS scheme and proved the scheme with *unconditional full anonymity* and *non-collusion*. Our scheme will be more effective in complex industry networks. Besides, the system can promote the social co-governance of food safety, which is essential to the food industry's sustainable development. The blockchain-cloud fusion scheme is a promising technique applied in democratic elections systems, online social networks, social co-governance in other industries, etc. [43, 50–56]

Data Availability

No data were used in this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2018YFC1604000) and the National Natural Science Foundation of China (Nos. 61572374 and U163620068).

References

[1] B. Tan, J. Yan, S. Chen, and X. Liu, "The impact of blockchain on food supply chain: the case of walmart," *Smart Blockchain*, pp. 167–177, 2018.

[2] N. Kshetri, "Blockchain and the economics of food safety," *IT Professional*, vol. 21, no. 3, pp. 63–66, 2019.

[3] R. Kamath, "Food traceability on blockchain: walmart's pork and mango pilots with IBM," *The Journal of the British Blockchain Association*, vol. 1, no. 1, pp. 47–58, 2018.

[4] L. M. Abenavoli, F. Cuzzupoli, V. Chiaravalloti et al., "Traceability system of olive oil: a case study based on the performance of a new software cloud," *Agronomy Research*, vol. 14, no. 4, pp. 1247–1256, 2016.

[5] C. V. Networking, "Cisco global cloud index: forecast and methodology, 2015-2020," 2016, https://www.iotjournaal.nl/wp-content/uploads/2017/02/white-paper-c11-738085.pdf.

[6] W. Zhang, M. Yang, X. Zhang, and H. Shi, "OMICC: an overlay multicast infrastructure based on cloud computing for streaming media data distribution," *ScienceAsia*, vol. 42S, no. 1, pp. 56–63, 2016.

[7] J. Wang, Z. Zhao, Z. Xu et al., "I-sieve: an inline high performance deduplication system used in cloud storage," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 17–27, 2015.

[8] W. Zhang, X. Zhang, and H. Shi, "MMCSACC: a multi-source multimedia conference system assisted by cloud computing for smart campus," *IEEE Access*, vol. 6, pp. 35879–35889, 2018.

[9] N. H. Ab Rahman and K.-K. R. Choo, "A survey of information security incident handling in the cloud," *Computers & Security*, vol. 49, pp. 45–69, 2015.

[10] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://www.researchgate.net/publication/228640975_Bitcoin_A_Peer-to-Peer_Electronic_Cash_System.

[11] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: a systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, p. 2337, 2018.

[12] H. J. P. Marvin, E. M. Janssen, Y. Bouzembrak, P. J. M. Hendriksen, and M. Staats, "Big data in food safety: an overview," *Critical Reviews in Food Science and Nutrition*, vol. 57, no. 11, pp. 2286–2295, 2016.

[13] S. Martins and Y. Yang, "Introduction to bitcoins: a pseudo-anonymous electronic currency system," in *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research (CASCON '11)*, pp. 349-350, IBM Corp., Berlin, Germany, 2011.

[14] S. Amit and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'05)*, pp. 457–473, Springer-Verlag, Berlin, Germany, 2005.

[15] Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011 (CT-RSA'11)*, pp. 376–392, Springer-Verlag, Berlin, Germany, 2011.

[16] J. Li and K. Kim, "Attribute-based ring signatures," p. 394, 2012, https://www.mendeley.com/catalogue/49cb6d99-ee43-3089-8dbd-946fcd75f7c8/.

[17] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *Lecture Notes in Computer Science*, M. Yung, Ed., vol. 2442, Berlin, Germany, Springer, 2002.

[18] X. Huang, Q. Tao, B. Qin, and Z. Liu, "Multi-authority attribute based encryption scheme with revocation," in *Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–5, IEEE, Berlin, Germany, 2015.

[19] M. Chase, "Multi-authority attribute based encryption," *Theory of Cryptography*, Springer, Berlin, Germany, pp. 515–534, 2007.

[20] K. Yang and X. Jia, "Attributed-based access control for multi-authority systems in cloud storage," in *Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, pp. 536–545, IEEE, Washington, DC, 2012.

[21] Z. Liu, Z. Cao, Q. Huang et al., "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," *Research in Computer Security*, Springer, Berlin, Germany, pp. 278–297, 2011.

[22] R. Pass and E. Shi, "FruitChains," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pp. 315–324, New York, NY, USA, 2017.

[23] L. Cocco, A. Pinna, and M. Marchesi, "Banking on blockchain: costs savings thanks to the blockchain technology," *Future Internet*, vol. 9, no. 3, p. 25, 2017.

[24] A. Maxmen, "AI researchers embrace Bitcoin technology to share medical data," *Nature*, vol. 555, no. 7696, pp. 293-294, 2018.

[25] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.

[26] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[27] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu, "A decentralizing attribute-based signature for healthcare blockchain," in *Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, IEEE, Las Vegas, NV, USA, 2018.

[28] A. N. Eltayieb, A. R. Elhabob, B. A. Hassan et al., "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud," *Journal of Systems Architecture*, vol. 102, Article ID 101653, 2020.

[29] M. Andoni, V. Robu, D. Flynn et al., "Blockchain technology in the energy sector: a systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[30] T. Bosona and G. Gebresenbet, "Food traceability as an integral part of logistics management in food and agricultural supply chain logistics management in food and agricultural supply chain," *Food Control*, vol. 33, pp. 32–48, 2013.

[31] B. Fahimnia, J. Sarkis, and H. Davarzani, "Green supply chain management: a review and bibliometric analysis," *International Journal of Production Economics*, vol. 162, pp. 101–114, 2015.

[32] Y.-P. Lin, J. Petway, J. Anthony et al., "Blockchain: the evolutionary next step for ICT e-agriculture," *Environments*, vol. 4, no. 3, p. 50, 2017.

[33] Q. Tao, X. Cui, X. Huang, A. M. Leigh, and H. Gu, "Food safety supervision system based on hierarchical multi-domain blockchain network," *IEEE Access*, vol. 7, pp. 51817–51826, 2019.

[34] K. A. Clauson, E. A. Breeden, C. Davidson et al., "Leveraging blockchain technology to enhance supply chain management in healthcare: an exploration of challenges and opportunities in the health supply chain," *Blockchain in Healthcare Today*, vol. 1, no. 3, pp. 1–12, 2018.

[35] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7081–7093, 2020.

[36] M. P. Caro, M. S. Ali, M. Vecchio et al., "Blockchain-based traceability in agri-food supply chain management: a practical implementation," in *Proceedings of the 2018 IoT vertical and topical summit on agriculture-tuscany (iot tuscany)*, pp. 1–4, IEEE, Tuscany, Italy, 2018.

[37] F. Antonucci, S. Figorilli, C. Costa, F. Pallottino, L. Raso, and P. Menesatti, "A review on blockchain applications in the agri-food sector," *Journal of the Science of Food and Agriculture*, vol. 99, no. 14, pp. 6129–6138, 2019.

[38] B. Amos, *Secure schemes for secret sharing and key distribution*, PhD Thesis, Isral institute of technology, Haifa, Israel, 1996.

[39] M. Castro, "Practical byzantine fault tolerance and proactive recovery," in *Proceedings of the Symposium on Operating Systems Design and Implementation*, pp. 173–186, USENIX Association, New Orleans, LA, USA, 1999.

[40] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, Springer, Berlin, Germany, 2011, http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=317E706FA7F1AD54460750309BC713F0?doi=10.1.1.

[41] A. L. Ferrara, M. Green, S. Hohenberger et al., "Practical short signature batch verification," in *Proceedings of the Cryptographers Track at the RSA Conference*, pp. 309–324, Springer, Berlin, Germany, 2009.

[42] G. Jens, O. Rafail, and S. Amit, "Perfect non-interactive zero knowledge for NP," in *Advances in Cryptology*Springer-Verlag, Berlin, Heidelberg, 2006.

[43] X. Boyen and B. Waters, "Compact group signatures without random oracles," in *Proceedings of theAnnual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 427–444, Springer, Berlin, Germany, 2006.

[44] L. Germany, "Blockchain data protection law | deloitte legal Deutschland [DB/OL], 2020-07-15," 2020, https://www2.deloitte.com/dl/en/pages/legal/articles/blockchain-datenschutzrecht.html.

[45] X. D. Liu, W. F. Zhang, and X. M. Wang, "Multi-Authority attribute-based alterable threshold ring signature without central authority," *Journal of Software*, vol. 29, no. 11, pp. 3528–3543, 2018.

[46] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the Annual International Cryptology Conference*, pp. 213–229, Springer, Berlin, Germany, 2001.

[47] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2014.

[48] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*, pp. 463–474, ACM, New York, NY, USA, 2013.

[49] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007.

[50] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Proceedings of the 2nd International Conference on Cryptology in Africa: Progress in Cryptology (AFRICACRYPT '09)*, pp. 198–216, Springer-Verlag, Berlin, Germany, 2009.

[51] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06*, Berlin, Germany,, 2006.

[52] B. Waters, "Ciphertext-Policy attribute-based encryption: an expressive, efficient, and provably secure realization," *Public Key Cryptography*, pp. 53–70, 2011.

[53] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 20–36, Springer, Berlin, Germany, 2008.

[54] G. Zyskind and O. Nathan, "Decentralizing privacy: using blockchain to protect personal data," in *Proceedings of the 2015 IEEE Security and Privacy Workshops*, pp. 180–184, IEEE, San Jose, CA, USA, 2015.

[55] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, San Jose, CA, USA, 2014.

[56] J. Sun, Y. Su, J. Qin, J. Hu, and J. Ma, "Outsourced decentralized multi-authority attribute based signature and its application in IoT," *IEEE Transactions on Cloud Computing*, p. 1, 2019.