Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 20, No. 2, 2026

VB-CP-ABE: An Improved CP-ABE Scheme Based on Blockchain Collaboration and Fine-rained Version Control

Guanglei Qiang *, Xiangyuan Qi and Zhuokun Fan

School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, China

* **Corresponding author**

Abstract: To address the shortcomings of data privacy protection in fruit traceability systems and the high revocation overhead of traditional CP-ABE schemes when personnel changes are frequent, this paper proposes an improved traceability system based on blockchain collaboration and version control. First, a two-layer "on-chain-off-chain" storage architecture is designed: IPFS is used to store AES-encrypted traceability data, while a consortium blockchain stores access policies, ciphertext indexes, and a global attribute version table, effectively alleviating on-chain storage pressure while ensuring data immutability. Second, a VB-CP-ABE algorithm based on version control is constructed. This algorithm cryptographically binds user attribute private keys to dynamic version numbers stored on the blockchain. When permission needs to be revoked, the version number of a specific attribute on the chain is updated via a smart contract, ensuring that the newly generated or published ciphertext cannot cancel the hash entries when decrypting with the old version private key, thus achieving immediate forward revocation. For users who still need to retain permissions, only incremental issuance of new version private key components for the affected attributes is required. Simulation results show that the proposed scheme outperforms comparative schemes in key generation, encryption, and decryption efficiency while ensuring access control security, demonstrating significant practical application value.

Keywords: Blockchain; Attribute-level encryption; IPFS; Traceability system; Fruit.

1. Introduction

With the continuous acceleration of the globalization of the fruit supply chain, fruits usually go through multiple regions and entities from planting, picking, processing to transportation and sales, and the supply chain structure is becoming increasingly complex. In recent years, food safety incidents have occurred frequently, and problems such as unclear product sources and difficulty in tracing the responsible parties have seriously threatened public health and market order. Against this background, building a reliable traceability system covering the entire process of fruit production, circulation and sales has become an urgent need to ensure food safety and promote the digital development of the agricultural supply chain [1].

Blockchain technology, with its decentralized, immutable and traceable characteristics, provides a new technical means for the construction of food traceability systems [2]. By recording key traceability information on the blockchain, it is possible to effectively prevent data from being maliciously tampered with and enhance the credibility of traceability results [3]. However, in practical applications, the fruit traceability system based on blockchain still faces many challenges. First, the contradiction between centralized storage and privacy protection is particularly prominent. Traditional traceability systems rely on centralized databases, which not only pose a single point of failure risk, but also make the data easy to be tampered with by internal personnel, lacking credibility. Although blockchain can alleviate the trust problem, its open and transparent characteristics make the data on the chain visible to all nodes. In the fruit supply chain, information such as logistics routes, purchase prices, and farmer identities are usually commercially sensitive data. If directly uploaded to the chain, privacy leaks will inevitably occur. Therefore, how to achieve fine-grained access control of sensitive traceability data while utilizing the immutable characteristics of blockchain has become an urgent problem to be solved [4].

Secondly, there is a natural conflict between the storage capacity of the blockchain and the scale of traceability data. Fruit traceability often involves large amounts of multimedia data such as pictures of the growing environment, quality inspection reports, and cold chain transportation videos. The blockchain adopts a full-node redundant storage mechanism, which has high storage costs and limited scalability. If all traceability data is written directly to the blockchain, the on-chain state will expand rapidly, seriously affecting the system throughput and operating efficiency. Therefore, how to design an "on-chain-off-chain collaborative" storage architecture to reduce the on-chain storage pressure while ensuring data integrity and verifiability is a key challenge in system architecture design [5].

To address the issues of data privacy and access control in the blockchain environment, the field of cryptography has proposed Ciphertext-Policy Attribute-Based Encryption (CP-ABE). CP-ABE is a public-key encryption mechanism that supports fine-grained access control. In this system, the data owner embeds the access control policy directly into the ciphertext when encrypting the data, and the users private key is associated with the set of attributes they possess. Data can only be successfully decrypted when the users attributes satisfy the access policy defined in the ciphertext [6]. Compared with traditional encryption methods based on identity or access control lists, CP-ABE moves the access control logic to the encryption stage, realizing a "encrypt first, then determine authority" security mode. It can support complex logical policy expressions (such as AND, OR, and threshold policies), and is particularly suitable for open data

sharing scenarios involving multiple roles and organizations. Therefore, CP-ABE is widely regarded as an ideal cryptographic tool for solving the data privacy protection problem in blockchain traceability systems.

However, in the dynamic fruit supply chain environment, the traditional CP-ABE scheme still has significant shortcomings in terms of the efficiency of permission revocation. The fruit supply chain is characterized by high personnel turnover and frequent entry and exit of nodes, such as temporary workers leaving and partners changing. In the existing CP-ABE scheme, once it is necessary to revoke the permission of a user or attribute, it is often necessary to regenerate the master key, update the private keys of all users whose permissions have not been revoked, or re-encrypt the historical ciphertext. This "one person revokes, everyone updates" mechanism brings computational and communication overhead that grows linearly with the number of users, causing the system to face efficiency bottlenecks in high-frequency permission change scenarios, making it difficult to meet the real-time and scalability requirements of practical applications.

In summary, the core issue this paper focuses on is how to construct a fruit supply chain traceability privacy protection system that can achieve efficient and secure data storage using blockchain and IPFS, while also enabling low-overhead, instant permission revocation through an improved CP-ABE mechanism.

2. Related work

Cited Policy Attribute-Based Encryption (CP-ABE) was first proposed by Bethencourt et al. This scheme embeds access control policies directly into the ciphertext, enabling data owners to flexibly define access conditions during the encryption phase, thereby achieving fine-grained data access control [7]. This scheme has an intuitive structure and strong expressive power, and is widely used in cloud storage and data sharing scenarios. However, it requires updating a large number of user keys or re-encrypting the ciphertext when attributes or users are revoked, resulting in a significant increase in system overhead with the scale of users, making it difficult to adapt to dynamic application environments.

To improve the security and practicality of CP-ABE, subsequent studies have made improvements in terms of efficiency and system architecture. Waters proposed a more efficient CP-ABE construction with provable security under the standard model, which enhanced the theoretical security basis of the scheme [8], but still did not fundamentally solve the problem of revocation cost caused by dynamic changes in permissions. In engineering applications, Green et al. proposed outsourcing the decryption calculation of CP-ABE to a semi-trusted server to reduce the computational burden on the user side [9]. This method improves the decryption efficiency to a certain extent, but does not reduce the system complexity of key update and management during the revocation process.

To address the issue of low revocation efficiency of CP-ABE in dynamic scenarios, Yu et al. proposed combining a proxy re-encryption mechanism to achieve attribute revocation. By delegating some re-encryption operations to proxy nodes, the computational burden on data owners and users is reduced [10]. This type of method can reduce the cost of direct re-encryption, but it introduces additional proxy trust assumptions, and the system structure is more complex. In recent years, some studies have further proposed server-assisted or time-sliced revocable ABE schemes to reduce revocation costs through periodic key updates or the participation of auxiliary nodes [11]. Although these methods theoretically improve revocation efficiency, in real systems with high personnel turnover, there are still problems such as high communication overhead and high implementation complexity.

In the context of blockchain applications, some studies have begun to try to combine CP-ABE with blockchain, using blockchain to record access policies, authorization status or audit information, while storing encrypted data in off-chain systems, thereby achieving secure data sharing in a decentralized environment [12][13]. Such solutions use blockchain to enhance the transparency and auditability of the system, but because on-chain data is publicly visible, access policies and attribute information may still cause privacy leaks. In addition, most solutions still rely on the key update mechanism of traditional CP-ABE when permissions are revoked, which is difficult to meet the real-time and scalability requirements of dynamic scenarios such as supply chains.

In summary, existing research has achieved significant results in fine-grained access control, system auditability, and security. However, it still falls short in providing permission revocation mechanisms that are low-overhead, require no re-encryption, and are suitable for dynamic supply chain scenarios. To address these shortcomings, this paper proposes a privacy protection scheme combining blockchain, IPFS, and an improved CP-ABE, using fruit supply chain traceability as an application scenario. The main contributions are as follows:

(1) A collaborative on-chain and off-chain fruit traceability system architecture is proposed. This architecture uses blockchain to store traceability indexes, access policies, and audit information, combined with IPFS storage for large-scale traceability data. This effectively reduces on-chain storage overhead while ensuring data integrity and verifiability.

(2) CP-ABE is introduced into the fruit supply chain traceability system to achieve fine-grained access control based on roles and business attributes, preventing the plaintext exposure of sensitive traceability data in the blockchain environment.

(3) A dynamic permission management and forward revocation mechanism based on attribute version control is designed. This prevents revoked users from decrypting the newly generated ciphertext after revocation, eliminating the need to re-encrypt existing ciphertext and significantly reducing system computation and communication overhead in scenarios with frequent personnel changes.

3. Overview of Attribute-Level Encryption Algorithms Based on Blockchain Collaboration and Fine-Grained Version Control

3.1. Overview of Multi-Entity Algorithms

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph). All manuscripts must be in English, also the table and figure texts, otherwise *sw* we cannot publish your paper. Please keep a second copy of your manuscript in your office.

The framework diagram of this system is shown in Figure 1. It consists of five entities: the Authorization Center (CA),

Data Owner (DO), Data User (DU), consortium blockchain nodes, and IPFS storage nodes, forming a distributed multi-entity cryptographic system.

The Authorization Center (CA) is responsible for initializing the systems public parameters and master key, acting as a bridge connecting off-chain identities and on-chain rights. During algorithm execution, the CA is responsible for executing the key generation algorithm. It must synchronize the attribute version table on the blockchain in real time, generating attribute private keys embedded with the latest version number for registered users, thereby establishing the timeliness of user identities. Furthermore, the CA is the sole initiator of permission revocation. When it is necessary to revoke a users permissions, the CA updates the version number of a specific attribute on the chain by calling the smart contract interface, triggering the invalidation of old private keys across the entire network using the version iteration mechanism, without interacting with the revoked user. For legitimate users who still need to retain permissions, the CA can incrementally issue new version attribute components as needed.

The Data Owner (DO) represents the data producers upstream in the fruit supply chain, such as planting bases, processing plants, or logistics companies. It is primarily responsible for data collection, encryption, and on-chain storage. The Data Provider (DO) first generates a random symmetric key locally to encrypt the original traceability data using AES to address the confidentiality issues of large data volumes. Then, the DO executes the encryption algorithm, retrieves the latest version of the current attribute from the blockchain, binds the access control policy to this version number, and encapsulates the symmetric key using CP-ABE. Finally, the DO uploads the encrypted data body to the distributed storage network and packages the metadata containing the policy, ciphertext, and index into a transaction, uploading it to the blockchain to complete data ownership confirmation and publication.

Data Users (DUs) are information consumers downstream in the supply chain, including regulatory bodies, end consumers, or retailers, responsible for data retrieval and decryption verification. DUs query and download metadata through the blockchain and simultaneously retrieve the encrypted data body based on the index. During the local decryption phase, the DU runs a decryption algorithm that compares the attribute version number carried in the ciphertext CT with the attribute version number embedded in the users private key SK. If the version s do not match, the denominator cannot eliminate the hash term in the numerator, resulting in decryption failure; if the versions match and the attribute set satisfies the access policy, the key material used for off-chain data decryption can be recovered. The Consortium Blockchain acts as an immutable "state machine" and "bulletin board" in the algorithm model, maintained collaboratively by multiple nodes. It not only stores the access policy matrix and ciphertext index, but more importantly, maintains a globally unique list of attribute versions. As the algorithms control layer, the blockchain uses smart contracts to constrain the atomicity of permission revocation operations, ensuring that any version update is traceable and irreversible. The version state on the blockchain serves as the benchmark for all encryption and decryption operations, guaranteeing the consistency of the permission view in a distributed environment.

The InterPlanetary File System (IPFS) forms the underlying data storage layer of the algorithm model, addressing the storage bottleneck faced by the blockchain. It receives large-volume fruit traceability data packets uploaded by data owners, encrypted with AES, and generates unique hash indexes (CIDs) using content addressing technology. By keeping the "heavy" data body off-chain and mapping only the "light" index hashes onto the chain, IPFS separates the control flow from the data flow, significantly reducing on-chain storage overhead and maintenance costs while ensuring data integrity and availability.

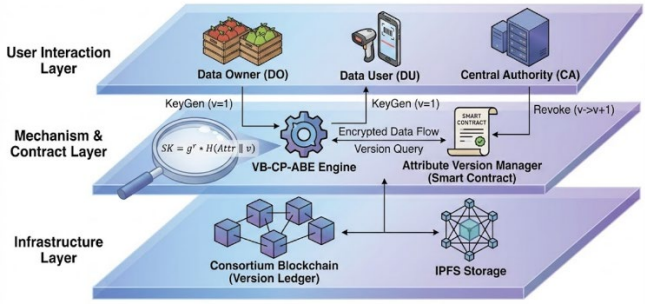

Fig. 1 VB-CP-ABE Algorithm Framework Diagram

3.2. Detailed system workflow

The system flowchart is shown in Figure 2. The symbols are explained in Table 1.

Table 1. Symbols and their meanings

Symbol	Meaning
CA, DO, DU	Authorization center, data owner, data user
CID	Content Addressing Hash
K_{sym}	Symmetric key
M	Original traceability data
C_{data}	Encrypted data
CT	Key Ciphertext
$\mathcal{V}$	Attribute version list
v_x	Version number
$H(attr \| v)$	Version binding hash
SK	User private key
$\mathbb{A}$	Access Policy

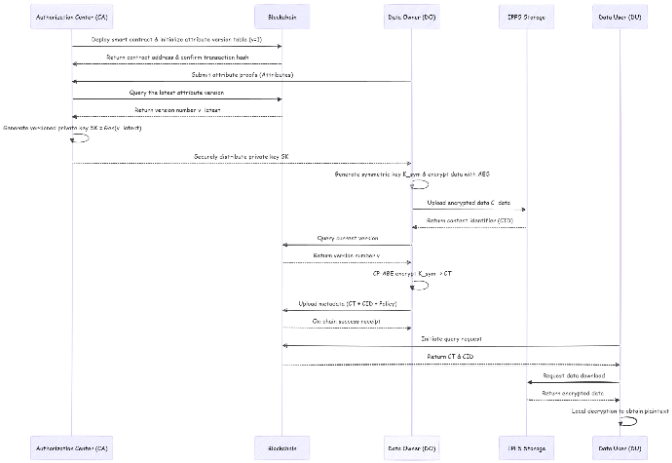

Fig. 2 System Flowchart

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph). All manuscripts must be in English, also the table

and figure texts, otherwise sw we cannot publish your paper. Please keep a second copy of your manuscript in your office.

(1) System Initialization Phase

$Setup(1^{\lambda}) \to (PP, MK, \mathcal{V}_0)$: The system initialization algorithm is executed by the Authorization Center (CA). Inputting the security parameter λ , the CA outputs the system public parameter PP and the system master key MK . Simultaneously, the CA initializes the attribute version table V_0 on the blockchain by calling a smart contract, setting the initial version number of all attributes to 1, thus establishing the systems trust baseline.

(2) Key Generation and Distribution Phase

$KeyGen(PP, MK, S, \mathsf{V}_{cur}) \to (SK)$: The user private key generation algorithm is executed by the CA. Input system public parameters PP , master key MK , user attribute set S , and the current attribute version table synchronized from the blockchain V_{cur} . The CA calculates and outputs the attribute private key SK , which embeds the current version information. This private key is sent to the data user (DU) via a secure channel; if the data owner (DO) also needs to access on-chain data as a data user, they are considered a DU and a private key is also issued to them.

(3) Data Encryption and Storage Stage

$DataEnc(M, K_{sym}) \to (C_{data}, CID)$: The data symmetric encryption algorithm is executed by the data owner (DO). The DO generates a random symmetric key K_{sym} locally and performs AES encryption on the original traceability data M to obtain the data ciphertext C_{data} . Then, C_{data} is uploaded to the IPFS distributed storage network and a unique content addressing hash CID is obtained. Then, $KeyEnc(PP, K_{sym}, \mathsf{A}, \mathsf{V}_{cur}) \to (CT)$ is executed to encapsulate the symmetric key. The DO converts the access policy A into an LSSS matrix (M, ρ) and selects a random vector $\vec{v} = (s, y_2, \ldots, y_n)$ to calculate the secret share $\lambda_i = M_i \cdot \vec{v}$ for each row. The algorithm calculates the ciphertext basic components $C = K_{sym} \cdot e(g,g)^{\alpha s}$ and $C' = g^{\beta s}$ to mask the symmetric key. For each row i of the matrix, the DO obtains the current version $v_{\rho(i)}$ of the corresponding attribute $\rho(i)$ from the blockchain and calculates the attribute ciphertext components $C_i = g^{\lambda_i}$ and $C_i' = H(\rho(i) \| v_{\rho(i)})^{\lambda_i}$. By embedding $H(\rho(i) \| v_{\rho(i)})$ in the ciphertext, the algorithm forces the decryptors private key to contain the same version hash item; otherwise, the pairing operation cannot be completed.

(4) Data Access and Decryption Phase

$Decrypt(PP, CT, SK) \to (K_{sym} / \perp)$: The decryption algorithm is executed locally by the data user (DU). Input public parameter PP , ciphertext CT obtained from the blockchain, and user private key SK . The algorithm first checks whether the version number in SK matches the version number in CT . If the versions match and the user attribute set S A satisfies the access policy, the algorithm outputs the symmetric key K_{sym} ; otherwise, it outputs $\perp$ (decryption failed).

$DataDec(C_{data}, K_{sym}) \to (M)$: The data recovery algorithm is executed by the DU. Input ciphertext C_{data} downloaded from IPFS and the decrypted symmetric key K_{sym} , and the DU outputs the original source plaintext M .

(5) Permission Revocation Phase

$Revoke(attr, \mathsf{V}_{old}) \to (\mathsf{V}_{new})$: The attribute revocation algorithm is executed by the CA through a smart contract. Input the attribute to be revoked $attr$ and the old version table V_{old} , the contract updates the on-chain state, and outputs the new version table V_{new} ($v_{attr} \leftarrow v_{attr} + 1$), thereby making the old version private key immediately invalid.

3.3. Construction of version-controlled CP-ABE algorithm

3.3.1. Setup

Performed by the Authorization Center (CA).

① Select the bilinear group: $\mathsf{G}_0, \mathsf{G}_T$ is a prime number of order p , with generator $g \in \mathsf{G}_0$ and pairing $e: \mathsf{G}_0 \times \mathsf{G}_0 \to \mathsf{G}_T$.

② Randomly select and calculate:

$$h = g^{\beta}, \quad f = g^{1/\beta}, \quad E = e(g,g)^{\alpha} \tag{1}$$

③ Define hash function $H: \{0,1\}^* \to \mathsf{G}_0$

④ On-chain initialization: Deploy the smart contract on the blockchain and initialize the attribute version table V . For all attributes in the system x , set the initial version $v_x = 1$.

⑤ Publish PP and secretly store MK :

$$PP = \{g, h, f, E, H\}, \quad MK = \{g^{\alpha}\} \tag{2}$$

3.3.2. Key Generation

① When a user registers, the CA executes algorithm $KeyGen$, inputting the user attribute set S .

② A unique identifier is randomly selected $r \in \mathbb{Z}_p$ (to prevent collusion attacks between different users).

③ The master part of the private key is calculated D :

$$D = (MK \cdot g^{r})^{1/\beta} = (g^{\alpha} \cdot g^{r})^{1/\beta} = g^{(\alpha + r)/\beta} \tag{3}$$

④ For each attribute j in attribute set S : Query the blockchain smart contract to obtain the current version number $v_j = \mathsf{V}[j]$ of attribute j . Randomly select $r_j \in \mathbb{Z}_p$. Calculate the attribute private key component $\{D_j, D_j'\}$:

$$D_j = g^{r} \cdot H(j \| v_j)^{r_j} \tag{4}$$

$$D_j' = g^{r_j} \tag{5}$$

⑤ Output the complete private key $SK = \{S, D, \forall j \in S : (D_j, D_j')\}$

3.3.3. Encryption and Storage Design

This solution employs a hybrid encryption and on-chain/off-chain collaborative storage strategy.

(1) Off-chain data encryption and storage

A random symmetric key $K_{sym} \in \mathsf{G}_T$ is generated by the data owner (DO). The original fruit traceability data M_{raw} is encrypted using the AES algorithm:

$$C_{data} = Enc_{AES}(K_{sym}, M_{raw}) \tag{6}$$

Upload C_{data} to IPFS to obtain index CID .

(2) On-chain Key Encapsulation

DO executes algorithm $Encrypt$ to encapsulate K_{sym}.

① Policy Transformation: The access policy A is transformed into an LSSS matrix (M,ρ), where M is the $l \times n$ matrix.

② Key Sharing: A random vector $\vec{v} = (s, y_2, \ldots, y_n) \in \square_p^n$ is selected, and the sharing share $\lambda_i = M_i \cdot \vec{v}$ for each row is calculated.

③ Version Acquisition: For each attribute $\rho(i)$ involved in the policy, DO queries the blockchain to obtain the latest version $v_{\rho(i)}$.

④ Ciphertext Calculation:

$$C = K_{sym} \cdot e(g,g)^{\alpha s}, \quad C' = h^s \tag{7}$$

$$C_i = g^{\lambda_i}, \quad C'_i = H(\rho(i) \parallel v_{\rho(i)})^{\lambda_i} \tag{8}$$

⑤ On-chain: DO submits transaction $TX = \{CID, \mathsf{A}, CT = (C, C', \{C_i, C'_i\})\}$ to blockchain.

3.3.4. Decryption

The $Decrypt$ algorithm is executed by the data user (DU). The decryption process includes two stages: version verification and key recovery.

(1) Pairing Operation and Version Check

Assume that the users attribute set S satisfies the access policy, and I is the set of matching row indices. For each $i \in I$, let the corresponding attribute be $x = \rho(i)$. The DU calculates the pairing components:

$$Node_i = \frac{e(C_i, D_x)}{e(C'_i, D'_x)} \tag{9}$$

Correctness derivation: Substitute the public key and ciphertext into the following:

$$Node_i = \frac{e(g^{\lambda_i}, g^{r \cdot H(x \| v_x)^{r_x})}}{e(H(x \| v_{\rho(i)})^{\lambda_i}, g^{r_x})} \tag{10}$$

If version v_x in the private key does not match version $v_{\rho(i)}$ in the ciphertext, then $H(x \square v_x) \neq H(x \square v_{\rho(i)})$, pairing cannot eliminate random numbers, and decryption fails. If they match, the bilinear property $e(u^a, v^b) = e(u,v)^{ab}$ is used to expand:

$$Node_i = \frac{e(g,g)^{r\lambda_i} \cdot e(g,H)^{r_x \lambda_i}}{e(H,g)^{\lambda_i r_x}} \tag{11}$$

The hash terms in the numerator and denominator cancel each other out, resulting in:

$$Node_i = e(g,g)^{r\lambda_i} \tag{12}$$

(2) Key Recovery

Using the LSSS linear reconstruction coefficients ω_i, calculate the aggregate value A:

$$A = \prod_{i \in I} (Node_i)^{\omega_i} = \prod_{i \in I} e(g,g)^{r\lambda_i \omega_i} = e(g,g)^{r \sum_{i \in I} \lambda_i \omega_i} = e(g,g)^{rs} \tag{13}$$

Calculate the intermediate term B:

$$B = \frac{e(C', D)}{A} \tag{14}$$

And: $e(C', D) = e(g^{\beta s}, g^{(\alpha + r)/\beta}) = e(g,g)^{s(\alpha + r)}$

Therefore

$$B = \frac{e(g,g)^{s(\alpha+r)}}{e(g,g)^{rs}} = e(g,g)^{\alpha s} \tag{15}$$

Finally, the symmetric key K_{sym} is recovered:

$$K_{sym} = \frac{C}{B} = \frac{K_{sym} \cdot e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} = K_{sym} \tag{16}$$

3.4. Dynamic permission management

To address the issue of high revocation overhead in traditional CP-ABE, this solution designs a blockchain-based dynamic revocation mechanism.

3.4.1. Attribute Cancellation

When an attribute needs to be revoked, the CA performs the following operations:

(1) On-chain update: The CA sends a transaction to invoke the smart contract.

(2) State change: The smart contract updates the stored version number to $v'_x = v_x + 1$.

(3) Revocation takes effect: When the DO reads v'_x, it uses $H(x \square v'_x)$ to generate ciphertext C'_i; the revoked user holds a private key D_x based on the old version v_x; during decryption, due to $H(x \square v_x) \neq H(x \square v'_x)$, the paired denominator cannot eliminate the hash item of the numerator, resulting in decryption failure. Revocation takes effect immediately on newly generated/newly published ciphertext after revocation; for old version ciphertext generated before revocation, if no ciphertext update is performed, this scheme does not guarantee that it will be immediately undecryptable.

3.4.2. Attribute authorization

When a user acquires a new attribute, the CA only needs to generate a corresponding new attribute private key component D_{new}, D'_{new} for that user based on the current on-chain version and issue it incrementally to the user, without needing to update the entire private key.

4. Experiment

This section analyzes and evaluates the actual performance of the proposed scheme through simulation experiments. The experimental platform is deployed on a computer equipped with an Intel(R) Core (TM) i7-10875H processor, 16 GB of memory, and a 512 GB solid-state drive, running Ubuntu 22.04 LTS (64-bit). The experimental software environment is built on Conda, using Python 3.8.20 as the primary development and runtime language. Cryptography-related experiments are implemented using the Charm-Crypto cryptographic framework, which integrates the GNU Multiple Precision Arithmetic Library (GMP 5.1.3) and the Pairing-Based Cryptography Library (PBC 0.5.14) to support large integer operations and bilinear pairing calculations. In the specific implementation, the SS512 elliptic curve parameter is selected, constructed on a supersingular symmetric elliptic curve of order 160 primes to complete the bilinear mapping operations required in the scheme.

To ensure the stability and repeatability of the experimental results, this paper uses the method of repeating the experiment multiple times and taking the average value for statistical analysis during the performance evaluation process. Specifically, under the same experimental parameter configuration, each group of experiments is independently repeated 30 times, and the arithmetic mean of each performance index: key generation time, encryption time and decryption time is taken as the final experimental result to reduce the influence of random factors on the experimental measurement results. To verify the effectiveness of the proposed scheme in terms of performance, this paper selects the BC-ABE-EF scheme [14] as the comparison object for experimental analysis.

4.1. Key generation time

Key generation (KeyGen) is one of the core operations in an attribute-based encryption system. Its main function is to generate corresponding attribute private keys for users based on their attribute sets by the authorization center. In practical fruit traceability systems, key generation operations typically occur during user registration or attribute update phases, and its efficiency directly affects the systems scalability in scenarios involving user growth or frequent permission changes. Experimental results are shown in Figure 3.

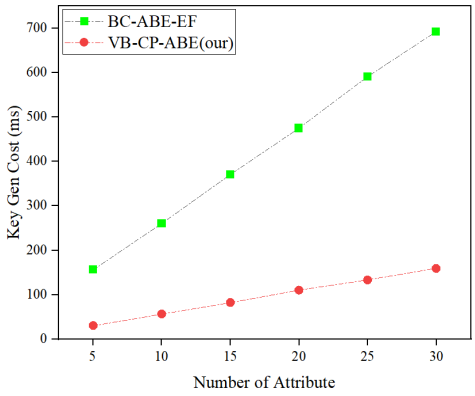

Fig. 3 Key generation time comparison

The figure compares the key generation time of the VB-CP-ABE scheme and the BC-ABE-EF scheme under different attribute numbers. As can be seen from the figure, the key generation time of both schemes exhibits an approximately linear growth trend with the increase in the number of attributes. This is because during the KeyGen process, each additional attribute requires the generation of a corresponding private key component, introducing additional exponential and hash calculation overhead. However, compared to the BC-ABE-EF scheme, VB-CP-ABE demonstrates a significantly lower key generation time across all test points. This is mainly due to the attribute versioning mechanism used in this scheme, which only needs to generate the corresponding attribute private key component for the current version during key generation, without introducing additional blockchain interaction or complex auxiliary parameter calculations. Experimental results show that when the number of attributes is 30, the key generation overhead of VB-CP-ABE is only a small fraction of that of the compared schemes, demonstrating its efficiency in the user registration and attribute update stages.

4.2. Key encryption time

The main function of the key encryption (Encrypt) stage is for the data owner to encapsulate the symmetric key using CP-ABE according to the access policy. This process directly determines the system response speed when data is published. In the fruit supply chain traceability scenario, the data owner may need to frequently publish new traceability data; therefore, encryption efficiency is an important indicator of system practicality. The experimental results are shown in Figure 4.

Figure 4 shows a comparison of the key encryption time of the two schemes under different numbers of attributes. It can be observed that the encryption time of both schemes increases linearly with the increase of the number of attributes in the access strategy. This is because in the CP-ABE encryption process, each strategy attribute corresponds to a row of LSSS matrices, requiring separate calculation of the ciphertext components. In contrast, the encryption time of VB-CP-ABE is consistently significantly lower than that of BC-ABE-EF. This is because the proposed scheme only introduces a lightweight version-binding hash operation during the encryption phase, without adding additional re-encryption or auxiliary calculation steps, thus avoiding complex on-chain collaboration or proxy computation overhead. Experimental results show that VB-CP-ABE can maintain low encryption latency even with continuously expanding attribute scale, making it suitable for deployment in practical traceability systems.

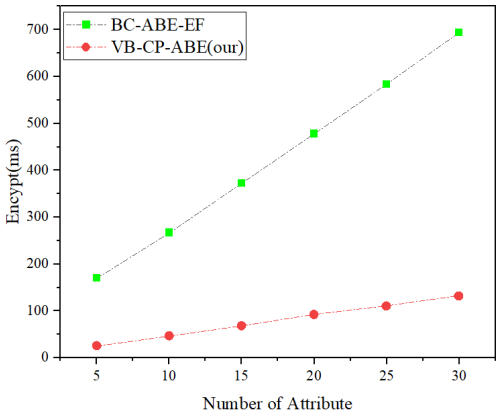

Fig. 4 Key encryption time comparison

4.3. Key decryption time

The key decryption phase is performed locally by the data user. Its main function is to verify whether the user attributes meet the access policy and recover the symmetric key. The performance of this phase directly affects the data access experience, especially in scenarios where regulatory authorities or consumers frequently query traceability information, where decryption efficiency is particularly critical. The experimental results are shown in Figure 5.

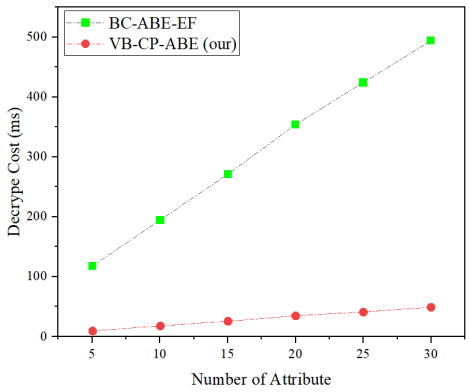

Fig. 5 Key decryption time comparison

The results in the figure show a comparison of decryption times for the two schemes under different attribute counts. As can be seen, the decryption time of the BC-ABE-EF scheme increases significantly with the number of attributes, while the decryption time of VB-CP-ABE shows a relatively gradual increase, remaining at a low level overall. The main reason for this difference is that VB-CP-ABE implements a fail-fast mechanism during the decryption phase through version number consistency checks: when the attribute version in the users private key does not match the version in the ciphertext, the algorithm can terminate the decryption process early in the pairing operation, thus avoiding

unnecessary computational overhead. Furthermore, for legitimate users, the decryption process does not require additional on-chain interaction or auxiliary decryption operations, further reducing decryption costs. Experimental results demonstrate that the proposed scheme has a significant advantage in decryption efficiency and can better meet the needs of high-concurrency access scenarios.

5. Summary

To address the shortcomings of privacy protection in fruit supply chain traceability systems and the high revocation overhead of traditional CP-ABE schemes in dynamic permission management scenarios, this paper proposes a VB-CP-ABE traceability privacy protection scheme based on blockchain collaboration and fine-grained version control. This scheme effectively alleviates the performance bottleneck of blockchain in large-scale traceability data storage by constructing an "on-chain-off-chain" collaborative storage architecture. Simultaneously, by combining attribute version control mechanisms with CP-ABE, it achieves forward instant permission revocation without re-encrypting existing ciphertext. Simulation results show that compared with the comparative scheme BC-ABE-EF, the proposed scheme has lower computational overhead in key operations such as key generation, key encryption, and key decryption, and maintains good scalability even with an increasing number of attributes. These characteristics make VB-CP-ABE particularly suitable for fruit supply chain traceability scenarios with frequent personnel movement and frequent permission changes. Future research will further consider introducing more flexible revocation granularity and a multi-authorization center mechanism while ensuring security, to improve the systems adaptability in larger -scale applications.

References

[1] Peng Z, Guanglei Q, Bo L. Fruit maturity detection based on improved YOLOv11 [J][J]. Modern Information Science and Technology, 2025, 9(08): 34-40.

[2] Yuan F, Zuo Z, Jiang Y, et al. AI-driven optimization of blockchain scalability, security, and privacy protection[J]. Algorithms, 2025, 18(5): 263.

[3] Zhao P, Qiang G, Lu B, et al. Practical Byzantine consensus algorithm based on Kademlia algorithm[C]//Proceedings of the 2025 9th International Conference on High Performance Compilation, Computing and Communications. 2025: 87-96.

[4] Alamsyah A, Widiyanesti S, Wulansari P, et al. Blockchain traceability model in the coffee industry[J]. Journal of Open Innovation: Technology, Market, and Complexity, 2023, 9(1): 100008.

[5] Qi X, Qiang G, Yuan F. Dairy Product Production Prediction Based on BiLSTM-Attention model[J]. Journal of Computer Science and Digital Technology, 2025, 1(1): 11-20.

[6] Das S, Namasudra S. MACPABE: Multi-Authority-based CP-ABE with efficient attribute revocation for IoT-enabled healthcare infrastructure[J]. International journal of network management, 2023, 33(3): e2200.

[7] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//2007 IEEE symposium on security and privacy (SP'07). IEEE, 2007: 321-334.

[8] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]//International workshop on public key cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 53-70.

[9] Green M, Hohenberger S, Waters B. Outsourcing the decryption of {ABE} ciphertexts[C]//20th USENIX security symposium (USENIX Security 11). 2011.

[10] Yu S, Wang C, Ren K, et al. Attribute based data sharing with attribute revocation[C]//Proceedings of the 5th ACM symposium on information, computer and communications security. 2010: 261-270.

[11] Penuelas-Angulo A, Feregrino-Uribe C, Morales-Sandoval M. Revocation in attribute-based encryption for fog-enabled internet of things: A systematic survey[J]. Internet of Things, 2023, 23: 100827.

[12] Liu C, Wang D, Li D, et al. Trusted access control mechanism for data with blockchain-assisted attribute encryption[J]. High-Confidence Computing, 2025, 5(2): 100265.

[13] Punia A, Gulia P, Gill N S, et al. A systematic review on blockchain-based access control systems in cloud environment[J]. Journal of Cloud Computing, 2024, 13(1): 146.

[14] Guo Y, Lu Z, Ge H, et al. Revocable blockchain-aided attribute-based encryption with escrow-free in cloud storage[J]. IEEE Transactions on Computers, 2023, 72(7): 1901-1912.