Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 20(7): 11957–11977.
DOI: 10.3934/mbe.2023531
Received: 26 February 2023
Revised: 09 April 2023
Accepted: 23 April 2023
Published: 12 May 2023

Research article

Blockchain-based multi-authority revocable data sharing scheme in smart grid

Xiao-Dong Yang[1,*], Ze-Fan Liao[1], Bin Shu[2] and Ai-Jia Chen[1]

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
[2] China Telecom WanWei Information Technology Co., LTD, Lanzhou 730030, China

* **Correspondence:** Email: y200888@163.com.

Abstract: In view of the problems of inefficient data encryption, non-support of malicious user revocation and data integrity checking in current smart grid data sharing schemes, this paper proposes a blockchain-based multi-authority revocable data sharing scheme in the smart grid. Using online/offline encryption technology with hybrid encryption technology enhances the encryption performance for the data owner. The use of user binary tree technology enables the traceability and revocability of malicious users. The introduction of multiple attribute authorization authorities eliminates the threat of collusive attacks that exist in traditional data-sharing schemes. In addition, the semi-honest problem of third-party servers is solved by uploading data verification credentials to the blockchain. The security analysis results show that the scheme can resist selective plaintext attacks and collusion attacks. The performance analysis results show that the proposed scheme has lower computational overhead and better functionality than similar schemes, which is suitable for secure data sharing in smart grids.

Keywords: smart grid; data sharing; hybrid encryption; block-chain; attribute-based encryption

1. Introduction

The smart grid is a new type of modern electric grid, which integrates energy technology and grid infrastructure with a high degree of integration of sensing and measurement technology, information and communication technology, analysis and decision-making technology and automatic control technology. In addition, it integrates the computing system and communication network's virtual environment with the physical environment of the power system to form a complex system that enables real-time sensing, dynamic changes and seamless information communication [1]. Digital twin is a technology that digitizes physical objects in the real world, models and simulates the various components of the power grid system and helps power grid operators achieve a comprehensive understanding and control of the grid. By combining digital twin technology, the smart grid can achieve more efficient, secure, and reliable power system operations [2–4]. This system is characterized by its user-friendly interface, high economic efficiency, exceptional reliability, compatibility and timely information updates. Users can use this system to access real-time information about electricity prices, electricity usage and other related information, making it easier to plan their energy usage. At the same time, smart grid terminals can continuously collect information about users' electricity usage and transmit it to the power company in real-time, allowing for real-time pricing and load balancing to ensure the stable operation of the grid system [5,6]. However, large-scale data collection undoubtedly raises concerns about user privacy and security. During the data collection process, there is a risk of leakage or tampering of critical information, such as user personal identification information or the power company's business secrets, which could potentially result in immeasurable losses for both the users and the power company [7–9]. Therefore, in the smart grid, data encryption protection and user access control have become critical for ensuring secure data sharing.

The ciphertext policy attribute-based encryption (CP-ABE) scheme associates user attributes with their encryption keys, allowing data owners to implement access control over their data while encrypting it, making it easier to achieve secure data sharing in the smart grid. Sahai et al. [10] first proposed the scheme of CP-ABE. Subsequently, data security sharing schemes based on CP-ABE and suitable for various real-life scenarios have been proposed one after another [11–16]. These schemes improve computational efficiency, security and attribute set size to varying degrees. However, these schemes do not consider user tracking and revocation, resistance to collusion attacks and how to achieve message integrity verification.

In traditional ciphertext policy attribute-based encryption schemes, in addition to storing the ciphertext, the cloud server not only stores the ciphertext but also holds the access policy. Therefore, all users who obtain the ciphertext can also obtain the access policy. However, an adversary can exploit the details in the access policy to deduce and obtain some private data. To prevent private data leakage through access policy, Zhang et al. [17] proposed a scheme to protect private data by hiding attribute values in access policy. However, Zhang's scheme is constructed by the composite order group, so its computation is less efficient, which is not suitable for the smart grid environment. Hui et al. [18] proposed a scheme for hiding attribute values in prime order groups, which achieves privacy protection for users while reducing computational overhead.

Moreover, some CP-ABE schemes are vulnerable to collusion attacks among users, in which users share their keys to obtain data. Therefore, tracing the malicious user who leaked the keys is also an important issue. Liu et al. [19,20] proposed a black-box and white-box tracking scheme to track malicious users. Only user tracking is insufficient for secure data sharing in the smart grid. An efficient user revocation mechanism is also necessary. Shi et al. [21–24] proposed different revocable CP-ABE schemes. Liu et al. [25] proposed a scheme that combines user tracking with revocation. Han et al. [26] further improved the scheme by incorporating privacy protection and enhancing computational efficiency. Then, Li et al. [27] proposed a traceable and revocable access control scheme which supports large attribute space, the decryption phase only requires constant-level bilinear pairing operations. Since a single attribute authority is not fully trusted and is not conducive to system expansion, Chase et al. [28] proposed a scheme for multiple attribute authorities. Later, De et al. [29] proposed a multi-authority CP-ABE scheme with higher encryption efficiency. Xiao et al. [30] proposed an access control scheme that incorporated revocation and multi-attribute authorities, but the scheme does not support user tracking. Sethi et al. [31] proposed a multi-authority scheme with revocable and traceable users. Datta et al. proposed a multi-authority scheme [32] based on asymmetric pairings; the scheme enhances security but is not efficient enough.

Online/offline encryption technology [33] preprocesses time-consuming operations such as bilinear pairing offline and completes them at idle time, so that users can simply perform simple operations online to get the data they need. This technology transfers the main computational overhead to the cloud server, greatly reducing the communication overhead and the computational overhead on the user. Hybrid encryption technology is to encrypt the primary data using a symmetric encryption algorithm such as AES and encrypt the key of the symmetric encryption algorithm using attribute encryption. The computational efficiency is further improved because the symmetric encryption algorithm has less overhead.

In real life, the cloud servers that store power grid data are semi-honest and curious, which poses potential risks to users' private data. With the development of blockchain-related technologies, the problem of semi-honest and curious third-party servers can be addressed using blockchain technology. Blockchain is a distributed ledger and database with the advantages of being tamper-proof, open and transparent [34]. The "transparency" feature of blockchain can solve the problem of information asymmetry between parties, enabling multiple entities to collaborate and trust and act in unison. The combination of blockchain and attribute-based encryption enhances the usability of encryption schemes.

To realize the secure sharing of grid data within the smart grid, we propose a blockchain-based multi-authority revocable data sharing scheme in the smart grid. The contributions can be summarized as follows:

1) The computational overhead of data owner and data user have been greatly reduced by introducing the online/offline encryption technology. The cloud server stores the conversion key to partially decrypt the ciphertext. Therefore, the data user requires a constant computation to decrypt the partially decrypted ciphertext. In addition, to reduce the communication overhead, we encrypt the grid data with symmetric encryption and encrypt the symmetric key with attribute-based encryption.

2) Data integrity verification. Based on the tamper-proof feature of the blockchain, the data owner can generate message verification credentials and upload them to the blockchain. The data user can download the verification credentials to compare with the received data to verify if the data has been tampered with.

3) User tracking and revocation. We introduce user binary tree and revocation list to manage users. The attribute authority can track a malicious user by the user's secret key, and then an attribute authority can add the user to the revocation list and update the ciphertext.

4) Privacy protection. Based on the scheme proposed in [26], user identity privacy is protected by combining it with policy hiding mechanism. By hiding the attribute values in the access policy, the adversary cannot infer user information from the access policy.

2. Preliminaries

2.1. Bilinear pairing

Let $\boldsymbol{G}$ and $\boldsymbol{G_T}$ be two multiplicative cyclic groups of prime order p, and g be a generator of $\boldsymbol{G}$. $\boldsymbol{G_T}$ has an efficient bilinear map $e:\boldsymbol{G}\times\boldsymbol{G}\to\boldsymbol{G_T}$ that satisfies the following three features:

1) Non-degeneracy: $e(g,g)\neq 1$.

2) Computability: For any element $M,N\in\boldsymbol{G}$, $e(M,N)$ could be efficiently computed by a polynomial-time algorithm.

3) Bilinearity: For any element $M,N\in\boldsymbol{G}$ and $a,b\in Z_p$, we can obtain $e(M^a,N^b)=e(M,N)^{ab}$.

2.2. Access structure

Let $P=\{P_1,P_2,...,P_n\}$ denote the set of n users. The collection $\widehat{A}$ is monotone for $\forall B,C$: if $B\in\widehat{A}$ and $B\subseteq C$, then $C\in\widehat{A}$. Let $\widehat{A}$ be a monotone nonempty subset of P, i.e., $\widehat{A}\subseteq 2^{\{P_1,P_2,...,P_n\}}\setminus\{\varnothing\}$, and then call $\widehat{A}$ a monotone access structure. The sets in $\widehat{A}$ are called authorized sets, and the rest of the sets are called the unauthorized sets.

2.3. User binary tree

Let U be the set of all users, RE be the revocation list and T_y be the user binary tree. T_y has the following features:

1) Every leaf node is related to a user u. Consider that $|U|$ denotes the number of users, and then there are $2|U|-1$ nodes in the T_y. Number these nodes by breadth-first traversal as $0\sim 2|U|-2$.

2) $path(\zeta)$ denotes the set of nodes on the path from the root node to the node ζ.

3) $cover(RE)$ denotes the minimal set of nodes for the users who are not included in the revocation list RE.

According to the above features of user binary tree, if a user is not in the revocation list RE, then there exists a unique node satisfying $\zeta_i{=}path(\zeta_u)\cap cover(RE)$.

Figure 1 is a simple example of user binary tree. If the revocation list RE is set to $RE=\{u_5,u_7\}=\{11,13\}$, then $cover(RE)=\{1,12,14\}$. The path of u_3 is $path(u_3)=\{0,1,4,9\}$. Therefore, the unique node $\zeta_i=cover(RE)\cap path(u_3)=\{1\}$.

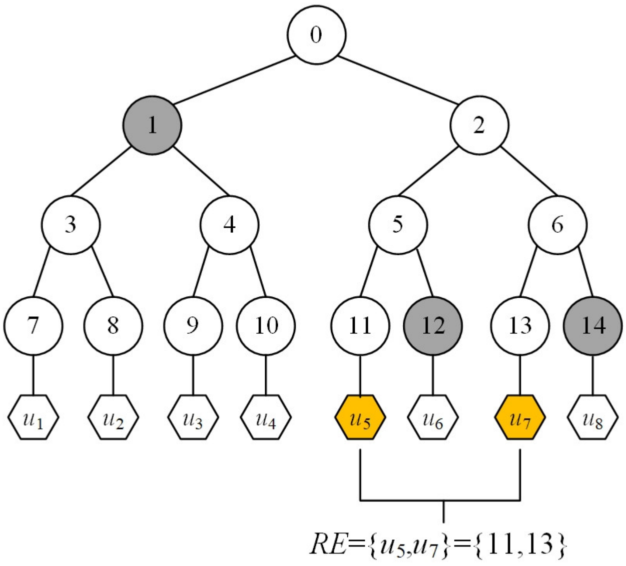

Figure 1. User binary tree.

2.4. Linear secret sharing scheme

Let $A^* = \{A_1^*, A_2^*, ..., A_n^*\}$ be the set of all attribute names, and each attribute name $A_i^* \in A^*$ corresponds to a set of attribute values $A_i^* = \{val_{i,1}, val_{i,2}, ..., val_{i,n_i}\}$, where n_i is the order of A_i^*. The access policy is denoted as $T = (M, \rho, V)$ in a linear secret sharing scheme (LSSS), where M is a matrix that has l row size and n column size. ρ is a function that maps every row of M into an attribute name in A^*. $V = \{v_{\rho(i)}\}_{i \in [1,l]}$ is the set of attribute values associated with (M, ρ). A LSSS includes the following two algorithms:

1) **Distribute:** For secret value $s \in Z_p$, randomly select a vector $f = (s, f_2, ..., f_n)^{\mathrm{T}}$, where $f_2, ..., f_n \in Z_p$. Calculate $\lambda_i = M_i \cdot f$, where M_i is the i^{th} row of matrix M. λ_i is a share of s that corresponds to $\rho(i)$.

2) **Reconstruct:** Let $S \in A^*$ be any authorized set, and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, ..., l\}$. Then, there exists a set of constants $\{\omega_i \in Z_p\}$ satisfying $\sum_{i \in I} \omega_i M_i = (1, 0, ..., 0)$. We could reconstruct the secret s by calculating $\sum_{i \in I} \omega_i \lambda_i = s$.

Let $\mathrm{S} = \{I_u, S\}$ be the set of user attributes and $\overline{T} = (M, \rho)$ be the access policy that hides the set of attribute values. $I_u \subseteq A^*$ is the set of user attribute names. $S = \{s_i\}_{i \in I_u}$ is the set of the user attribute values. For $\forall i \in I$, where $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, ..., l\}$, if i satisfies (M, ρ) and $s_{\rho(i)} = v_{\rho(i)}$, then we say that S matches T.

2.5. Complex assumption

Definition 1. (q-BDHE assumption)

Let $\boldsymbol{G}$ and $\boldsymbol{G}_T$ be two multiplicative cyclic groups of prime order p and g be a generator of $\boldsymbol{G}$. $\boldsymbol{G}_T$ has an efficient bilinear map $e : \boldsymbol{G} \times \boldsymbol{G} \to \boldsymbol{G}_T$. Randomly select $t, f \in Z_p$ and compute vector $J=(g,g^{t},g^{f},g^{f^{2}},...,g^{f^{q}},g^{f^{q+2}},...,g^{f^{2q}})$. Then, the $q-\text{BDHE}$ assumption is defined as follows: There is no polynomial-time algorithm that can distinguish between $e(g,g)^{f^{q+1}t}\in \boldsymbol{G}_T$ and $B\in \boldsymbol{G}_T$ by a non-negligible advantage ε.

3. System and secure models

3.1. System model

The system model consists of the following five entities:

1) Data Owner (DO): DO collects grid data from users' houses and specifies an access policy to encrypt grid data, then generates verification credential and uploads it to the blockchain.

2) Cloud Server (CS): CS stores the ciphertext and the verification credentials while generating the transformed ciphertext for DU. CS is semi-honest.

3) Data User (DU): DU obtains ciphertext from CS. DU can decrypt ciphertext only if DU's attributes satisfy the access policy. There may be malicious users in DU.

4) Attribute Authority (AA): AA generates public parameters and private keys. AA cannot access user secret keys or user identity information alone.

5) Blockchain: Blockchain stores and transmits public parameters and verification credentials, preventing them from being tampered with.

The system model is shown in Figure 2.

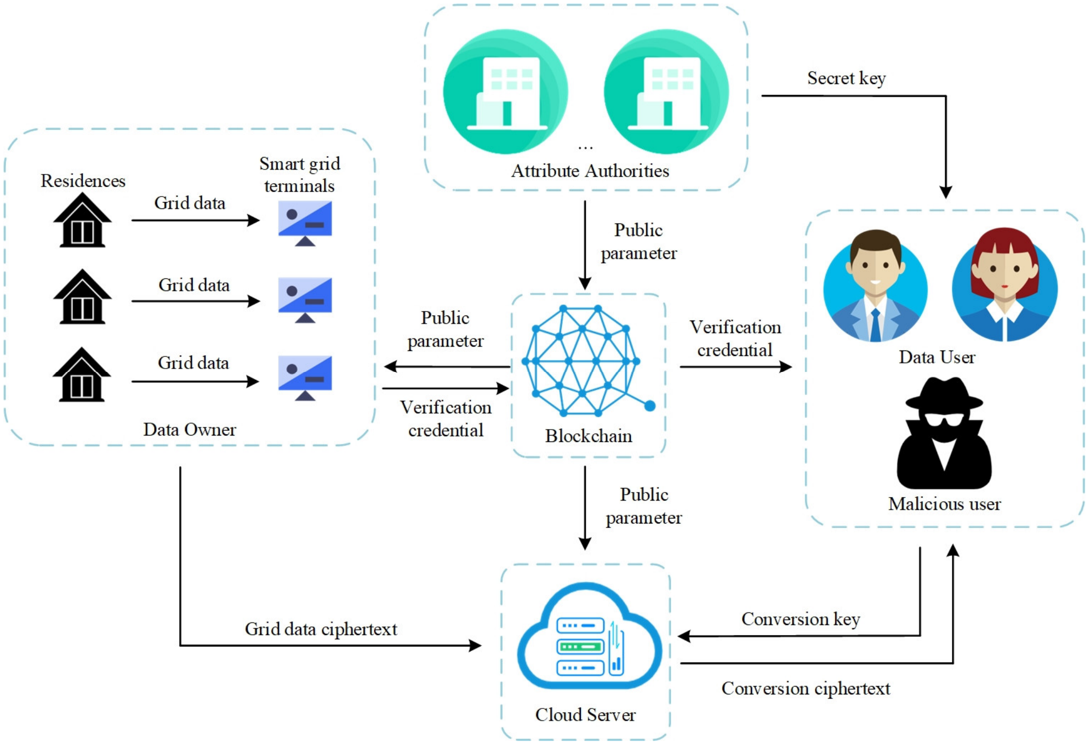

Figure 2. System model.

3.2. Formal definition

The proposed scheme consists of nine algorithms, and we describe each algorithm as follows:

1) **Setup:** AA generates public parameter PP and master key MSK, and then AA publishes public parameter PP to other entities.

2) **KeyGen:** AA runs the algorithm and inputs user's attribute set $\mathrm{S}=\{I_u,S\}$; AA outputs user's secret key SK.

3) **CKeyGen:** DU runs the algorithm. DU randomly chooses recover key HK_u and then generates conversion key CK_u according to HK_u.

4) **Enc.offline:** DO executes the algorithm. DO outputs intermediate ciphertext IT according to the public parameter PP.

5) **Enc.online:** DO executes the algorithm. DO inputs grid data MSG, revocation list RE, public parameter PP and outputs ciphertext CT.

6) **Conversion:** CS executes the algorithm. DU inputs ciphertext CT, conversion key CK_u, public parameter PP. Then, CS outputs conversion ciphertext $\widehat{CT}$.

7) **Decrypt:** DU executes the algorithm. DU inputs conversion ciphertext $\widehat{CT}$, secret key SK, recover key HK_u, public parameter PP and outputs the decrypted and verified grid data $MSG^{'}$.

8) **Track:** AA runs the algorithm. First, check if the secret key SK satisfies the required conditions for tracking. If SK is satisfied, the user identity is calculated based on the public parameter PP, the conversion ciphertext $\widehat{CT}$ and the secret key SK. Eventually, AA gets the tracked user u and the updated revocation list $RE^{'}$.

9) **Update:** AA and CS run this algorithm. AA inputs the updated revocation list $RE^{'}$, and CS outputs the updated ciphertext $CT^{'}$.

The system flow chart is shown in Figure 3.

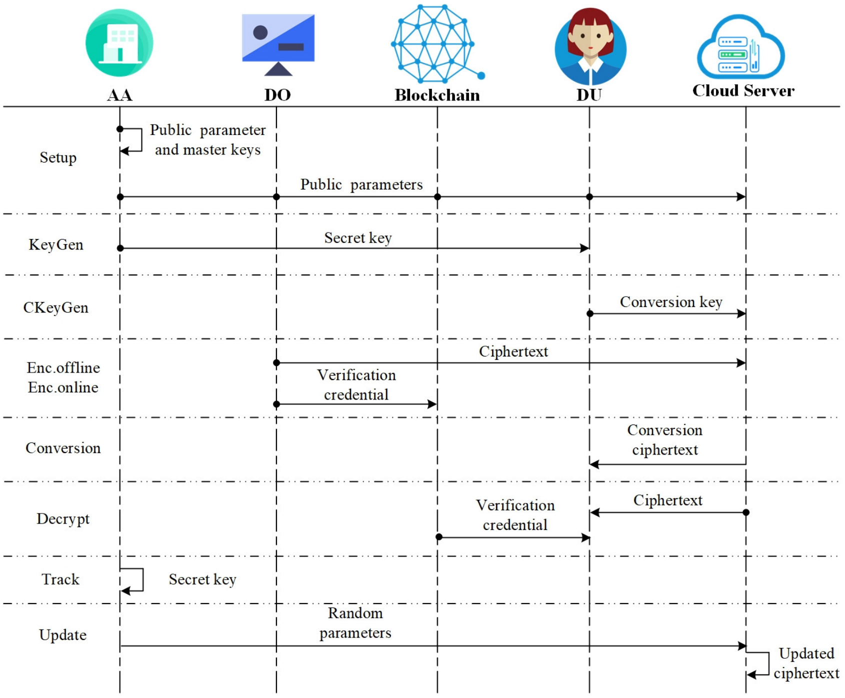

Figure 3. System flow chart.

3.3. Security model

The indistinguishability under chosen-plaintext attack (IND-CPA) of the proposed scheme can be described as a security game between a challenger C and an adversary A. The procedures are as follows:

Setup: A selects and submits an access policy $T^* = \{M^*, \rho^*, V\}$ and a user revocation list RE^* to C, where M is a matrix that has l row size and n column size. ρ is a function that maps every row of M into an attribute name in A^*. C runs **Setup** algorithm in Section 3.2 and generates public parameter, then sends public parameter to A.

Phase 1: A asks C for the decryption key of attribute sets $(u_1, \mathrm{S}_1)(u_2, \mathrm{S}_2)...(u_q, \mathrm{S}_q)$, where q means the number of users.

If $\mathrm{S}_i \in T^*$ and $u_i \notin RE^*$, which indicates the user's attribute sets satisfy the access policy, and the user is not listed in the revocation list, then abort.

If $\mathrm{S}_i \notin T^*$ or $u_i \in RE^*$, which indicates the user's attribute sets do not satisfy the access policy, or the user is listed in the revocation list, then C generates a decryption key and sends it to A.

Challenge: A submits two messages m_0 and m_1 of the same length. Next, C randomly chooses $\varpi \in \{0,1\}$, then encrypts message m_ϖ under access policy $T^* = \{M^*, \rho^*, V\}$ and revocation list RE^*. Eventually, C sends ciphertext CT^* as a challenge to A.

Phase 2: Phase 2 remains the same as the operation of Phase 1.

Guess Phase: A outputs a guess ϖ' about ϖ and wins the game if $\varpi = \varpi'$. The advantage of A winning the security game can be described as: $\varepsilon = \left|\Pr[\varpi = \varpi'] - 1/2\right|$.

Definition 2: In polynomial time, if the adversary A cannot win the above security game with non-negligible advantage, then the proposed scheme is indistinguishable under chosen-plaintext attacks.

4. Scheme construction

4.1. Setup

In this section, the attribute authorities generate the public parameter PP and the master key, then publish the public parameters to other entities.

The scheme contains N attribute authorities $\{AA_1, ..., AA_N\}$. Attribute authorities generate public parameter PP as follows:

1) Attribute authority AA_1 selects two multiplicative cyclic groups $\boldsymbol{G}$ and $\boldsymbol{G}_T$ of prime order p. $\boldsymbol{G}_T$ has an efficient bilinear pairing which can be described as $e: \boldsymbol{G} \times \boldsymbol{G} \to \boldsymbol{G}_T$, and g is a generator of $\boldsymbol{G}$. AA_1 randomly selects $m, n \in G$. E_f and D_f are symmetric encryption and decryption algorithms.

2) AA_1 selects collision-resistant hash functions $H_0(): G \to \{0,1\}^*$, $H_1(): \{0,1\}^* \to \{0,1\}^{\ell_{H_1}}$ and secure key generation function H.

3) Each attribute authority randomly chooses $k_{f,j} \in G$, then sends it over a secure channel to other attribute authorities ($j \in \{1, 2, ..., N\}$). Each attribute authority computes and gets the symmetric key $k_f = H(\prod_{j=1}^{N} k_{f,j})$ used to encrypt the user's identity.

4) For each node in T_y, AA_1 randomly selects $\delta_i \in Z_p$ to get $\{\delta_i\}_{i=0}^{2|U|-2}$ and calculates

$\{\psi_i = g^{\delta_i}\}_{i=0}^{2|U|-2}$. AA_1 sends $\{\delta_i\}_{i=0}^{2|U|-2}$ over a secure channel to other attribute authorities. Each leaf node in T_y is associated with a user u and assigned a unique value $v_u \in Z_p$.

5) Each attribute authority randomly chooses $\alpha_j, a_j \in Z_p$ and calculates $Y_j = e(g,g)^{\alpha_j}$, g^{a_j}. Then, it sends Y_j and g^{a_j} to other attribute authorities. After receiving Y_j and g^{a_j} from other attribute authorities, each attribute authority calculates

$$Y = \prod_{j=1}^{N} e(g,g)^{\alpha_j} = e(g,g)^{\sum_{j=1}^{N} \alpha_j}; \tag{1}$$

$$\prod_{j=1}^{N} g^{a_j} = g^{\sum_{j=1}^{N} a_j}. \tag{2}$$

For ease of presentation, $\widetilde{A}$ and A are used below to represent $\sum_{j=1}^{N} \alpha_j$ and $\sum_{j=1}^{N} a_j$.

6) Attribute authorities publicize the system parameter $PP = \{p, G, G_T, e, g, m, n, Y, A, \{\psi_i\}_{i=0}^{2|U|-2}, E_f, D_f, H_0(), H_1(), H()\}$ and upload PP to the blockchain. Each attribute authority saves its secret key $MSK_j = \{a_j, \alpha_j, \{\delta_i\}_{i=0}^{2|U|-2}\}$.

4.2. KeyGen

In this section, the attribute authorities generate the secret key SK for data users.

$\mathrm{S} = \{I_u, S\}$ is the set of user attributes, where $I_u \in A^*$ is the attribute name, and $S = \{s_i\}_{i \in I_u}$ is the set of attribute values. Attribute authorities generate the secret key SK as follows:

1) Each attribute authority calculates $E_{id} = E_f(k_f, v_u)$.

2) Every attribute authority randomly selects $r_j \in Z_p$, then calculates $D_1 = g^{Ar_j}, D_2 = g^{r_j}$, $D_3 = E_{id}$, $K_j^{'} = m^{r_j}$ and sends to other attribute authorities.

3) Let ζ_u be the leaf node associated with the user u in the binary tree and the set of leaf nodes be $path(\zeta_u) = \{\zeta_0, \zeta_1 ..., \zeta_u\}$. ζ_0 is the root node. Each attribute authority calculates $D_0 = g^{r_j / \delta_{\zeta_u}}$ and sends to other attribute authorities.

After receiving components from other attribute authorities, each attribute authority calculates secret key as follows:

$$\widetilde{D_1} = \prod_{j=1}^{N} g^{Ar_j} = g^{A\sum_{j=1}^{N} r_j}; \tag{3}$$

$$\widetilde{D_2} = \prod_{j=1}^{N} g^{r_j} = g^{\sum_{j=1}^{N} r_j}; \tag{4}$$

$$K = g^{\frac{\widetilde{A}}{A+E_{id}}} \cdot \prod_{j=1}^{N} K_j^{'} = g^{\frac{\widetilde{A}}{A+E_{id}}} \cdot m^{\sum_{j=1}^{N} r_j}; \tag{5}$$

$$\widetilde{D_\theta} = \prod_{j=1}^{N} g^{s_i r_j} n^{-(A+E_{id}) r_j} = g^{s_i \sum_{j=1}^{N} r_j} \cdot n^{-(A+E_{id}) \sum_{j=1}^{N} r_j}; \tag{6}$$

$$\widetilde{D_0} = \prod_{j=1}^{N} g^{\frac{r_j}{\delta_{\zeta_u}}} = g^{\frac{\sum_{j=1}^{N} r_j}{\delta_{\zeta_u}}}. \tag{7}$$

4) The secret key $SK=\{\widetilde{D_0},\widetilde{D_1},\widetilde{D_2},D_3,\{\delta_{\zeta}\}_{\zeta\in path(\zeta_u)},K,\{\widetilde{D_\theta}\}_{\theta\in I_s},\mathrm{S}\}$. R is used below to represent $\sum_{j=1}^{N}r_j$.

4.3. CKeyGen

In this section, the data user generates the recover key and the conversion key, then sends these keys to the cloud server.

The data user randomly selects $x_u\in Z_p$ as recover key HK_u and calculates the conversion key CK_u as follows:

$$L_1=\widetilde{D_1}^{1/x_u} \tag{8}$$

$$L_2=\widetilde{D_2}^{1/x_u} \tag{9}$$

$$L_3=D_3 \tag{10}$$

$$L_4=K^{1/x_u} \tag{11}$$

The conversion key $CK_u=\{L_1,L_2,L_3,L_4,L_\theta\}$.

4.4. Enc.offline

In this section, the data owner completes time-consuming operations such as bilinear pairing offline to reduce the communication overhead, while generating the symmetric key.

The data owner sets access policy $T=(M,\rho,V)$ and generates intermediate ciphertext as follows:

1) Randomly select $R_u\in G$ and secret value $s\in Z_p$, and calculate symmetric key $k_u=H(R_u)$. Then, calculate $C=R_u e(g,g)^{\widetilde{A}s}$, $C_0=g^s$, $C_0^{'}=g^{\widetilde{A}s}$ and $\{\psi_i^{\,s}\}_{i=0}^{2|U|-2}$.

2) For b^{th} row M_b in the matrix M, randomly select $z_b,q_b,t_b\in Z_p$, where $b=1,2,...,l$. The data owner calculates $C_{b,1}=m^{z_b}n^{q_b}$, $C_{b,2}=g^{-q_b t_b}g^{z_b}$ and $C_{b,3}=g^{q_b}$.

3) The intermediate ciphertext $IT=\{s,C,C_0,C_0^{'},R_u,k_u,\{\psi_i^{\,s}\}_{i=0}^{2|U|-2},\{z_b,q_b,t_b,C_{b,1},C_{b,2},C_{b,3}\}_{b=1,2,...,l}\}$.

4.5. Enc.online

In this section, the data owner encrypts the grid data using the symmetric key generated in Section 4.4 and then encrypts the symmetric key using the attribute-based encryption algorithm.

The data owner executes the following processes to encrypt symmetry key and grid data:

1) Calculate the verification credential $Token=H_1(H_0(R_u)\,||\,MSG)$ and send it to the blockchain. Then use the symmetric key k_u to encrypt grid data MSG and obtain grid data ciphertext CT_M .

2) Randomly select a vector $\kappa=(s,f_2,...,f_n)^{\mathrm{T}}$. For each row M_b in the matrix M, calculate $z_b^{'}=M_b\kappa$.

3) Calculate $C_{b,4}=z_b^{'}-z_b$ and $C_{b,5}=q_b(v_{\rho(b)}-t_b)$, and then get the minimal set $cover(RE)$

through the binary tree T_y. Next, select the cipher component $\{F_i = \psi_i{}^s\}_{i \in cover(RE)}$ according to $cover(RE)$.

4) The ciphertext $CT = \{\overline{T}, RE, CT_M, C, C_0, C_0{}', \{C_{b,1}, C_{b,2}, C_{b,3}, C_{b,4}, C_{b,5}\}_{b \in 1,2,...,l}, \{F_i\}_{i \in cover(RE)}\}$, where $\overline{T}$ is the access policy of hidden attribute values.

4.6. Conversion

In this section, the cloud server preprocesses the ciphertext data and sends the conversion ciphertext to the data user.

The cloud server executes the following processes to generate conversion ciphertext $\widehat{CT}$:

1) Check if user u is included in the revocation list RE. If the user is included in RE, it means the user is illegal, and then terminate the process. Otherwise, according to the attributes set uploaded by the user, get the set of rows $I_r = \{b : \rho(b) \in S\} \subseteq \{1,2,...,l\}$ and the set of constants $\{\omega_b \in Z_p\}_{b \in I_r}$. When $\{z_b{}'\}$ is valid, $\sum_{b \in I_r} \omega_b z_b{}' = s$.

2) If the value of attribute $s_{\rho(b)}$ in the user's secret key SK satisfies the value $v_{\rho(b)}$ in the access policy, calculate as follows:

$$
\begin{aligned}
Q_1 &= e(L_2{}^{L_3} L_1, C_{b,1} m^{C_{b,4}}) \cdot e(L_2, C_{b,2} g^{C_{b,4} - C_{b,5}}) \cdot e(L_{\rho(i)}, C_{b,3}) \\
&= e(g^{(A+E_{id})R/x_u}, m^{z_b{}'} \cdot n^{q_b}) \cdot e(g^{R/x_u}, g^{z_b{}' - q_b \cdot v_{\rho(i)}}) \cdot e(g^{s_i R/x_u} \cdot n^{-(A+E_{id})R/x_u}, g^{q_b}) \\
&= e(g,m)^{(A+E_{id}) z_b{}' R/x_u} e(g,g)^{z_b{}' R/x_u}
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
Q_2 &= \prod_{b \in I_c} (Q_1)^{\omega_b} \\
&= e(g,m)^{(A+E_{id}) sR/x_u} e(g,g)^{sR/x_u}
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
Q_3 &= e(K, C_0{}^{L_3} C_0{}') \\
&= e(g^{\widetilde{A}/[(A+E_{id})x_u]} m^{R/x_u}, g^{(A+E_{id})s}) \\
&= e(g,g)^{s\widetilde{A}/x_u} e(g,m)^{(A+E_{id}) sR/x_u}
\end{aligned}
\tag{14}
$$

$$
C_t = \frac{Q_3}{Q_2} = \frac{e(g,g)^{s\widetilde{A}/x_u}}{e(g,g)^{sR/x_u}}
\tag{15}
$$

3) The cloud server sends the conversion ciphertext $\widehat{CT} = \{RE, C_t, C, \{F_i\}_{i \in cover(RE)}\}$ to the user.

4.7. Decrypt

In this section, the data user decrypts the conversion ciphertext and gets the symmetric key. Then the data user uses the symmetric decryption algorithm to get the grid data ciphertext and completes the data integrity verification using the verification credential downloaded from the blockchain.

The data user u executes the algorithm to decrypt ciphertext as follows:

1) If u is not included in the revocation list RE, then there exists a unique node $\zeta_i = path(\zeta_u) \cap cover(RE)$, where ζ_u is the leaf node associated with u.

2) According to the node ζ_i and user's secret key $\{\widetilde{D_0}, \{\delta_\zeta\}_{\zeta \in \mathrm{path}(\zeta_u)}\}$, get δ_{ζ_u} and δ_{ζ_i}, and then calculate $o = \delta_{\zeta_u} / \delta_{\zeta_i}$.

3) Calculate the components as follows:

$$Q_4 = e(\widetilde{D_0}, F_i)^o = e(g^{R/\delta_{\zeta_u}}, (g^{\delta_{\zeta_i}})^s)^{\delta_{\zeta_u}/\delta_{\zeta_i}} = e(g,g)^{Rs} \tag{16}$$

$$R_u' = \frac{C}{(C_t)^{x_u} Q_4} = \frac{C}{e(g,g)^{\tilde{A}s}} \tag{17}$$

4) The symmetric key $k_u' = H_0(R_u')$. Use k_u' to decrypt grid ciphertext, then obtain the grid data $MSG' = D_f(k_u', CT')$.

5) Download the verification credential $Token$ from the blockchain, and if the equation $Token = H_1(H_0(R_u') \| MSG')$ holds, then the grid data MSG' is valid.

4.8. Track

In this section, the attribute authority checks the user's secret key SK and tracks the user with the binary tree. Then, the attribute authority adds the user to the revocation list RE and updates the revocation list.

The attribute authority executes the algorithm to trace users as follows:

1) Check whether the user's secret key SK satisfies the following conditions:

a) $D_3 \in Z_p$, $\widetilde{D_0}, \widetilde{D_1}, \widetilde{D_2}, K, \widetilde{D_\theta} \in G$

b) $e(g, D_1) = e(g^A, D_2) \neq 1$

c) $e(K, g^A g^{D_3}) = e(g,g)^{\tilde{A}} e(\widetilde{D_2}^{D_3} \cdot \widetilde{D_1}, m) \neq 1$

d) $\exists \theta \in I_s, s.t.\ e(\widetilde{D_\theta}, g) e(\widetilde{D_2}^{D_3} \cdot \widetilde{D_1}, n) = e(\widetilde{D_2}.g)^{s_\theta} \neq 1$

2) If the user secret key SK passes the above check, then calculate $v_u = D_s(k_f, D_3)$. Next, use v_u to find the corresponding user u in the binary tree and add u to the revocation list. $RE' = RE \cup \{u\}$ is the updated revocation list.

4.9. Update

In this section, the attribute authorities and the cloud server complete the ciphertext update after user revocation based on the updated nodes set $cover(RE')$.

The attribute authority and the cloud server execute the algorithm as follows together:

1) Each attribute authority randomly selects $\varphi_j \notin Z_p$ and sends it to other attribute authorities. Then, each attribute authority calculates $\tilde{\varphi} = \sum_{j=1}^{N} \varphi_j$ and $\delta' = \{\tilde{\varphi} \cdot \delta_i\}_{i=0}^{2|U|-2}$ and sends $\tilde{\varphi}$ and δ' to the cloud server.

2) The cloud server obtains new nodes set $cover(RE')$ from RE'. For node $\zeta_{i'} \in cover(RE')$, there exist two cases when updating the ciphertext:

a) For the revocation list RE before the update, there exists a node $\zeta_i \in cover(RE)$ that makes $\zeta_{i'} = \zeta_i$ hold. Then, set $F_{i'} = F_i$.

b) For the revocation list RE before the update, there exists a node $\zeta_i \in cover(RE)$ such that ζ_i is an ancestor of $\zeta_{i'}$. Suppose that $path(\zeta_{i'}) = path(\zeta_i) \cup \{\zeta_{i+1}, ..., \zeta_{i'}\}$, let $P_i = \psi_i$ and calculate iteratively as follows, where $k = i+1, ..., i'$:

$$P_{k+1} = (P_k)^{\frac{\psi_{k+1}'}{\psi_k'}} = \psi_{k+1}^s \tag{18}$$

Next, set $F_{i'} = P_{i'}$.

Eventually, the updated ciphertext is $CT' = \{\overline{T}, RE', CT_M, C, C_0, C_0', \{C_{b,1}, C_{b,2}, C_{b,3}, C_{b,4}, C_{b,5}\}_{b \in 1,2,...,l}, \{F_{i'}\}_{i' \in cover(RE')}\}$

5. Security analysis

5.1. Collusion Attack Resistance

Theorem 1. The proposed scheme can resist the collusion attack by $N-1$ attribute authorities if the discrete logarithm hardness assumption (Discrete Logarithm Problem, DLP) holds.

Proof: Each attribute authority randomly selects $a_j \in Z_p$ and sends g^{a_j} to other attribute authorities. From the discrete logarithmic difficulty problem, it is difficult for an adversary to infer a_j from g^{a_j}. Thus, even if there exist $N-2$ attribute authorities in collusion with the adversary, there will still be a parameter that cannot be determined, and the adversary cannot guess valid g^A. Then, the adversary cannot construct a valid master key. Therefore, the proposed scheme can resist the collusion attack by $N-1$ attribute authorities.

5.2. Security proof

Theorem 2. The proposed scheme has indistinguishability under chosen-plaintext attack (IND-CPA) if $q-\mathrm{BDHE}$ hardness assumption holds, where $q > 2|U|-2$.

Proof: Assume an adversary A can break the proposed scheme with a non-negligible advantage ε, and then there exists a challenger C that can solve the $q-\mathrm{BDHE}$ problem with the advantage $\varepsilon/2$. C executes algorithms as follows:

Let $\boldsymbol{G}$ and $\boldsymbol{G_T}$ be two multiplicative cyclic groups of prime order p and g be a generator of $\boldsymbol{G}$. $\boldsymbol{G_T}$ has an efficient bilinear map $e: \boldsymbol{G} \times \boldsymbol{G} \to \boldsymbol{G_T}$. C randomly selects $\mu \in \{0,1\}$. Give a vector $J = (g, g^t, g^f, g^{f^2}, ..., g^{f^q}, g^{f^{q+2}}, ..., g^{f^{2q}})$, and if $\mu = 1$, then C calculates $Z = e(g,g)^{f^{q+1}t}$. Otherwise, C randomly selects $Z \in \boldsymbol{G_T}$.

Setup: A selects and submits an access policy $T^* = \{M^*, \rho^*, V\}$ and a user revocation list RE^* to C, where M is a matrix that has l row size and n column size. ρ is a function that maps every row of M into an attribute name in A^*. C randomly selects $\sigma', a \in Z_p$ and sets $e(g,g)^\alpha = e(g,g)^{\sigma'} e(g^t, g^{t^q})$, then calculates g^a, $m = g^f$, $n = g^{f^q}$, where $\alpha = \sigma' + t^{q+1}$.

For the revocation list RE^*, C sets $I_{RE^*} = \{\zeta_i \in path(u) \mid u \in RE^*\}$ and randomly selects

$\{d_i \in Z_p\}_{i\in\{0,2|U|-1\}}$. If $\zeta_i \in I_{RE^*}$, set $\psi_i = g^{d_i} g^{f^i}$, and then $\delta_i = d_i + f^i$. Otherwise, set $\psi_i = g^{d_i} g^{f^q}$, and then $\delta_i = d_i + f^q$.

C sends the public parameter $PP = \{p, \boldsymbol{G}, \boldsymbol{G_T}, e, g, m, n, e(g,g)^{\alpha}, g^{a}, \{\psi_i\}_{i=0}^{2|U|-2}\}$ to A.

Phase 1: A submits user attribute sets $u, \mathrm{S} = \{I_u, S\}$ to require corresponding secret keys. $I_u \in A$ is user's attribute name. $S = \{s_i\}_{i\in I_u}$ is user's attribute value set. For every attribute value s_j in attribute space and $\forall i \in \{1,2,...,l^*\}$, if $s_j = v_{\rho^*(i)}$, it means that s_j satisfies the attribute value in the access policy, and set $\tau_j = s_j + \sum_{h=1}^{n^*} f^h M_{j,h}^*$. Otherwise, set $\tau_j = s_j$. If $u \notin RE^*$, then set $\delta_i = d_{\zeta_i} + f^q$. If $u \in RE^*$, then for $\forall \zeta_i \in path(u)$, there exists $\zeta_i \in I_{RE^*}$, and set $\delta_i = d_{\zeta_i} + f^{\zeta_i}$. If $\mathrm{S} \in T^*$, randomly select $\iota \in Z_p$ and calculate $t = [f^{q-1}/(a+\iota)] \cdot (M_{i,1}^* / M_{i,2}^*) + [-f^q/(a+\iota)]$. Otherwise, randomly select vector $\vec{b} = (b_1, b_2, ..., b_{n^*}) \in Z_p$, where $b_1 = -1$, and all i in $\rho^*(i) \in I_u$ satisfy $M_i^* \cdot \vec{b} = 0$. Then, randomly select $r, \iota \in Z_p$ and calculate $t = (r + b_1 f^q + \cdots + b_{n*} \cdot f^{q-n^*+1})/(a+\iota)$.

Challenge: A submits two messages m_0 and m_1 of the same length to C. C randomly selects $\varpi \in \{0,1\}$ and computes $C = m_c e(g,g)^{\alpha t}$, $C_0 = g^t$, $C_0^{'} = g^{\alpha t}$. Next, C randomly selects $k_2, ..., k_{n^*} \in Z_p$ and compute the vector $\vec{h} = (t, tf + k_2, ..., tf^{n^*-1} + k_{n^*})$. Then C computes as follows:

$$C_{i,1} = \prod_{j=2}^{n^*} (g^{fk_j})^{M_{i,j}^*} \prod_{j=1}^{n^*} (g^{tf^j})^{M_{i,j}^*} g^{-af^{q+i}} \tag{19}$$

$$C_{i,2} = (g^{v_{\rho^*(i)}})^{-af^i} \prod_{j=2}^{n^*} (g^{f^j M_{i,j}^*})^{-af^i} \cdot \prod_{j=2}^{n^*} (g^{k^j})^{M_{i,j}^*} g^{-af^{q+i}} \cdot \prod_{j=2}^{n^*} (g^{-af^{q+i}})^{M_{i,j}^*} \tag{20}$$

$$C_{i,3} = g^{-af^i} \tag{21}$$

For $\forall \zeta \in path(u)$, due to $\zeta \notin I_{RE^*}$, C can obtain $\psi_i = g^{d_i} g^{f^q}$ and $\delta_i = d_i + f^q$. Finally, C sends the ciphertext $CT^* = \{C, C_0, C_0^{'}, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[1,l^*]}, \{F_i\}_{i\in cover(RE^*)}\}$ to A.

Phase 2: Phase 2 remains the same as the operation of Phase 1.

Guess Phase: A outputs a guess $\varpi^{'}$ about ϖ. If $\varpi = \varpi^{'}$, C outputs the guess $\mu^{'} = 1$ about μ, and then A obtains the normally generated ciphertext. The advantage that A obtains the normally generated ciphertext is $\varepsilon = \Pr[\varpi = \varpi^{'} \mid \mu = 1] - (1/2)$, and it means $\Pr[\varpi = \varpi^{'} \mid \mu = 1] = \Pr[\mu = \mu^{'} \mid \mu = 1]$. Therefore, the probability that C wins the game is $\Pr[\mu = \mu^{'} \mid \mu = 1] = \varepsilon + (1/2)$. Otherwise, C outputs the guess $\mu^{'} = 0$ about μ, A obtains a randomly selected element from $\boldsymbol{G_T}$, so that A cannot obtain any information from ϖ, $\Pr[\varpi = \varpi^{'} \mid \mu = 1] = 1/2$. Then $\Pr[\varpi \neq \varpi^{'} \mid \mu = 0] = \Pr[\mu = \mu^{'} \mid \mu = 0]$ can be concluded. Thus, the probability that C wins the game is $\Pr[\mu = \mu^{'} \mid \mu = 0] = 1/2$.

Eventually, if C can solve the $q-$BDHE problem, then the advantage of C can be described as follows:

$$\Pr[\mu=\mu^{'}]=\Pr[\mu=\mu^{'}\mid\mu=1]\cdot\Pr[\mu=1]+\Pr[\mu=\mu^{'}\mid\mu=0]\cdot\Pr[\mu=0]=(\varepsilon+\frac{1}{2})\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2}=\frac{\varepsilon}{2} \tag{22}$$

6. Performance

In this section, we compare several existing schemes [26,27,29,30,31] with our scheme. Table 1 defines the symbols that appear below.

Table 1. The implication of symbols.

Symbols	Implications
E_z	Exponential operations on groups G, G_T
P_z	Bilinear pairing operations
M_z	Multiplication operations in groups G, G_T
s_p	The number of attributes in the access policy
s_c	The number of attributes in secret key that satisfy the access policy
R	The length of $cover(RE)$

The hardware and software environments for implementing the experiment are as follows: A laptop equipped with 3.2GHz AMD Ryzen 7 6800H CPU and 8GB RAM was used, and the operating system is 64-bit Windows 11. We used JPBC 2.0.0 library for the experiment. The prime-order bilinear pairing is constructed on the 160-bit elliptic curve group, which is based on the curve $y^2=x^3+x$. Table 2 lists the runtime of the three cryptographic operations in the above environment. Table 3 compares the proposed scheme with previous ABE schemes in terms of the number of attribute authorities, verification, revocation, policy hiding, ability to encrypt offline, and whether blockchain technology is introduced. Table 4 shows the performance of the proposed scheme versus other schemes in the encryption and other phases. Figures 4 to 7 present the variation of the computation time as the user attribute space grows.

Table 2. Cryptographic operation runtime.

Cryptographic operation	P_z	E_z	M_z
Time (ms)	8.1	6.2	0.002

Table 3. Functionality comparisons.

Scheme	Attribute Authority	Verification	Hidden policy	Blockchain	Revocation	Outsource decryption
[26]	Single	×	√	×	√	×
[27]	Single	×	×	×	√	×
[29]	Multiple	×	×	×	×	√
[30]	Multiple	×	×	√	√	×
[31]	Multiple	×	×	×	√	√
Ours	Multiple	√	√	√	√	√

Table 3 presents an analysis of various schemes. Specifically, the scheme [26] provides traceability, while the scheme presented in [27] adds policy hiding and revocation. However, both schemes rely on a single attribute authority and thus fail to resist collusion attacks from malicious users and attribute authorities. On the other hand, the scheme proposed in [29] adopts multiple attribute authorities to resist collusion attacks and employs online/offline encryption to enhance efficiency. While the schemes in [30,31] support flexible revocation, they suffer from high computational overhead. However, the above schemes lack support for integrity verification. In contrast, the proposed scheme integrates revocation, hidden policy, multiple attribute authorities and online/offline encryption. Additionally, the proposed scheme incorporates verification and blockchain technology to ensure data integrity.

Table 4. Computational overhead comparison.

Scheme	Encryption		Decryption	Tracking	CTUpdate
	Offline	Online			
[26]	-	$(s_p+1)M_z+$ $(3+4s_p+R)E_z$	$(1+4s_c)P_z+$ $(3+3s_c)M_z+$ $(3+s_c)E_z$	$(6+s_p)P_z+$ $(3+s_p)M_z+$ $(3+s_p)E_z$	RE_z
[27]	-	$(R+8+2s_p)M_z$	$4P_z+$ $(s_c-R+2)E_z$	$(3s_p)P_z+$ s_pM_z+ $(3s_p)E_z$	P_z+ $(3+2R)E_z$
[29]	$(8s_p+1)E_z$	$(3s_p+1)M_z$	$P_z+3M_z+2E_z$	-	-
[30]	-	$(s_p+1)E_z$	s_cP_z+ $(2s_c+1)E_z$	-	-
[31]	-	$P_z+(3s_p)M_z+$ $(5s_p)E_z$	$4P_z+4M_z+2E_z$	$(1+4s_p)P_z+$ $(2+4s_p)M_z+$ $(1+3s_p)E_z$	P_z+ $(1+8s_p)M_z+$ $(1+8s_p)E_z$
Ours	$(5s_p+2)E_z$	$(2s_p+R)M_z$	$P_z+2M_z+2E_z$	$(5+s_p)P_z+$ $(3+s_p)M_z+$ $(2+s_p)E_z$	RE_z

Table 4 shows the computational overhead of various schemes. From Table 4, we can observe that the encryption time of [27,29] and the proposed scheme is much less than other schemes. Furthermore, the proposed scheme attains a constant level of computational overhead in the decryption and update phase. In the tracking phase, both the proposed scheme and the other schemes contain a large number of complex operations, so the computational overhead is approximated.

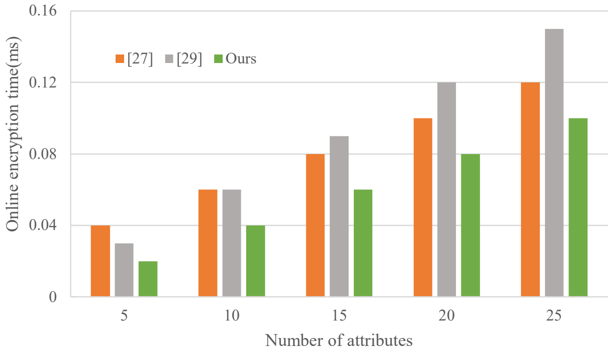

Figure 4. Online encryption time.

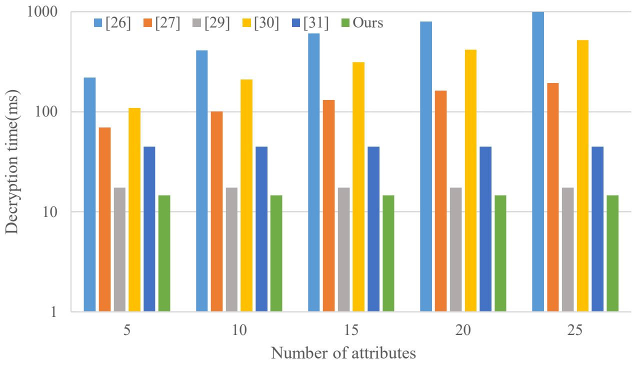

Figure 5. Decryption time.

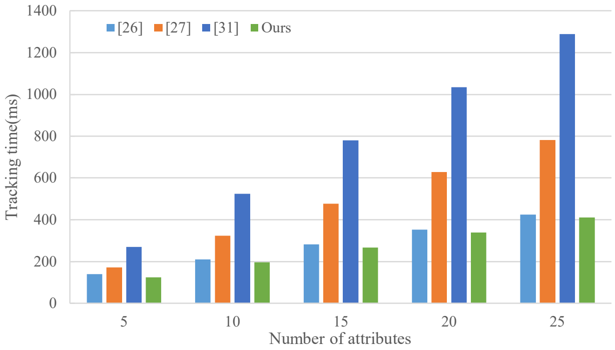

Figure 6. Tracking time.

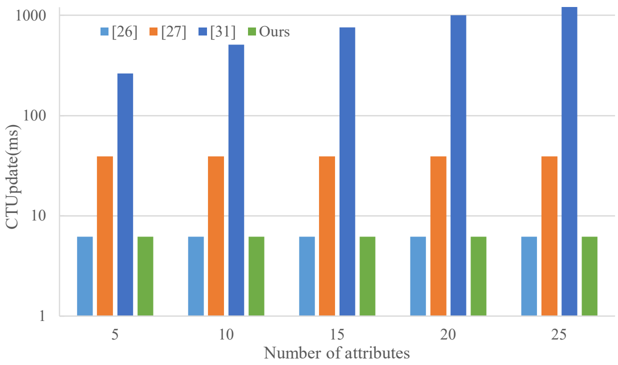

Figure 7. CTUpdate time.

Figure 4 describes the change in online encryption time as the attribute space grows. We can find that the computation time for encryption grows linearly as the attribute space grows, and the encryption overhead is reduced because the proposed scheme uses online offline encryption to pre-process the data.

Figure 5 illustrates the decryption time variation. The schemes [26,27,30] do not consider the case of large attribute space, and the computational overhead of these schemes increases rapidly as the attribute number grows. The scheme [29,31] and the proposed scheme outsource the complex calculations such as bilinear pairing to the cloud server. Therefore, the proposed scheme achieves constant-level decryption time.

Figure 6 shows the tracking time of various schemes. Due to more bilinear pairing operations, the scheme [31] takes more time than other schemes. Even with the realization of data integrity verification at the same time, the tracking time of the proposed scheme remains at a low growth rate.

Figure 7 shows the ciphertext update time. The ciphertext update times for the proposed scheme and the scheme [26] are only related to the number of revoked users. Therefore, the update time is much lower than other schemes. From the indicated results, the proposed scheme is more efficient and suitable for smart grid, while achieving the desired security goals. In summary, the experimental results are consistent with the above theoretical analysis in Table 4.

7. Conclusions

To address the important issues in the existing CP-ABE schemes, such as the inability to resist collusion attacks, high computation overhead and lack of support for outsourced verification, this paper proposes a blockchain-based multi-authority revocable data sharing scheme. The proposed scheme can resist collusion attacks and realize access control with users' attribute sets. Moreover, the proposed scheme can trace malicious users and then revoke the user by using the binary tree. The proposed scheme introduces online/offline encryption and hybrid encryption, so that the computation overhead of the decryption phase is not affected by user attributes. Thus, the proposed scheme supports large attribute space and is more suitable for secure data sharing in multi-user smart grid. We also presented security proof to demonstrate that the proposed scheme is IND-CPA-secure under the hardness assumptions of $q-\text{BDHE}$.

Acknowledgments

The authors appreciate the editors and reviewers for their valuable comments and insightful suggestions. This work was supported by National Natural Science Foundation of China (No. 61662069), the China Postdoctoral Science Foundation (No. 2017M610817).

Conflict of interest

All authors declare there is no conflict of interest.

References

1. Y. Tang, Q. Wang, M. Ni, Y. Liang, Analysis of cyber attacks in cyber physical power system, *Autom. Electr. Power Syst.*, **40** (2016), 148–151. http://dx.doi.org/10.7500/AEPS20160123101
2. H. Gong, S. Cheng, Z. Chen, Q. Li, Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics, *Nucl. Sci. Eng.*, **196** (2022), 668–693. https://doi.org/10.1080/00295639.2021.2014752
3. P. T. Baboli, D. Babazadeh, D. R. K. Bowatte, Measurement-based modeling of smart grid dynamics: a digital twin approach, in *2020 10th Smart Grid Conference (SGC)*, Kashan, (2020), 1–6. https://doi.org/10.1109/SGC52076.2020.9335750
4. H. Gong, S. Cheng, Z. Chen, Q. Li, C. Quilodrán-Casas, D. Xiao, et al., An efficient digital twin based on machine learning SVD autoencoder and generalised latent assimilation for nuclear reactor physics, *Ann. Nucl. Energy*, **179** (2022), 109431. https://doi.org/10.1016/j.anucene.2022.109431
5. J. Gao, Y. Xiao, J. Liu, W. Liang, C. L. P. Chen, A survey of communication/networking in smart grids, *Future Gener. Comput. Syst.*, **28** (2012), 391–404. https://doi.org/10.1016/j.future.2011.04.014
6. B. Lu, Y. Ma, Research on communication system of advanced metering infrastructure for smart grid and its data security measures, *Power Syst. Technol.*, **37** (2013), 2244–2249.
7. S. R. Rajagopalan, L. Sankar, S. Mohajer, H. V. Poor, Smart meter privacy: a utility-privacy: framework, in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, (2011), 190–195. https://doi.org/10.1109/SmartGridComm.2011.6102315
8. H. Li, X. Liang, R. Lu, X. Lin, X. Shen, EDR: an efficient demand response scheme for achieving forward secrecy in smart grid, in *2012 IEEE Global Communications Conference (GLOBECOM)*, (2012), 929–934. https://doi.org/10.1109/GLOCOM.2012.6503232
9. L. Sankar, S. Kars, R. Tandon, H. V. Poor, Competitive privacy in the smart grid: an information-theoretic approach, in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, (2011), 220–225. https://doi.org/10.1109/SmartGridComm.2011.6102322
10. A. Sahai, B. Waters, Fuzzy identity-based encryption, in *Advances in Cryptology – EUROCRYPT 2005*, Springer, Berlin, Heidelberg, (2005), 457–473. https://doi.org/10.1007/11426639_27

11. M. Joshi, K. Joshi, T. Finin, Attribute based encryption for secure access to cloud based EHR systems, in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, (2018), 932–935. https://doi.org/10.1109/CLOUD.2018.00139
12. Z. Liu, L. Jiang, X. Wang, S. M. Yiu, Practical attribute-based encryption: outsourcing decryption, attribute revocation and policy updating, *J. Network Comput. Appl.*, **108** (2018), 112–123. https://doi.org/10.1016/j.jnca.2018.01.016
13. M. Cui, D. Han, J. Wang, An efficient and safe road condition monitoring authentication scheme based on fog computing, *IEEE Internet Things J.*, **6** (2019), 9076–9084. https://doi.org/10.1109/JIOT.2019.2927497
14. Y. Rouselakis, 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*, Berlin, (2013), 463–474. https://doi.org/10.1145/2508859.2516672
15. W. Fan, L. Li, X. Chen, H. Jiang, Z. Li, K. C. Li, Deploying parallelized ciphertext policy attributed-based encryption in clouds, *Int. J. Comput. Sci. Eng.*, **16** (2018), 321–333. https://doi.org/10.1504/IJCSE.2018.091784
16. X. Li, K. Liang, Z. Liu, D. Wong, Attribute based encryption: traitor tracing, revocation and fully security on prime order groups, in *Proceedings of the 7th International Conference on Cloud Computing and Services Science - CLOSER*, (2017), 309–320. https://doi.org/10.5220/0006220203090320
17. Y. Zhang, D. Zheng, R. H. Deng, Security and privacy in smart health: efficient policy-hiding attribute-based access control, *IEEE Internet Things J.*, **5** (2018), 2130–2145. https://doi.org/10.1109/JIOT.2018.2825289
18. H. Cui, R. H. Deng, J. Lai, X. Yi, S. Nepal, An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited, *Comput. Networks*, **133** (2018), 157–165. https://doi.org/10.1016/j.comnet.2018.01.034
19. Z. Liu, Z. Cao, D. S. Wong, Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay, in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, Berlin, (2018), 475–486. https://doi.org/10.1145/2508859.2516683
20. Z. Liu, X. Wang, L. Cui, Z. L. Jiang, C. Zhang, White-box traceable dynamic attribute-based encryption, in *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, (2017), 526–530. https://doi.org/10.1109/SPAC.2017.8304334
21. Y. Shi, Q. Zheng, J. Liu, Z. Han, Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation, *Inf. Sci.*, **295** (2015), 221–231. https://doi.org/10.1016/j.ins.2014.10.020
22. V. H. Hoang, E. Lehtihet, Y. Ghamri-Doudane, Forward-secure data outsourcing based on revocable attribute-based encryption, in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, (2019), 1839–1846. https://doi.org/10.1109/IWCMC.2019.8766674
23. G. Xiang, B. Li, X. Fu, M. Xia, W. Ke, An attribute revocable CP-ABE scheme, *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, (2019), 198–203. https://doi.org/10.1109/CBD.2019.00044

24. S. Wang, K. Guo, Y. Zhang, Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage, *PLoS One*, **13** (2018), e0206952. https://doi.org/10.1371/journal.pone.0206952
25. Z. Liu, S. Duan, P. Zhou, B. Wang, Traceable-then-revocable ciphertext-policy attribute-based encryption scheme, *Future Gener. Comput. Syst.*, **93** (2019), 903–913. https://doi.org/10.1016/j.future.2017.09.045
26. D. Han, N. Pan, K. Li, A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection, *IEEE Trans. Dependable Secure Comput.*, **19** (2020), 316–327. https://doi.org/10.1109/TDSC.2020.2977646
27. Q. Li, B. Xia, H. Huang, Y. Zhang, TRAC: traceable and revocable access control scheme for mHealth in 5G-enabled IIoT, *IEEE Trans. Ind. Inf.*, **18** (2022), 3437–3448. https://doi.org/10.1109/TII.2021.3109090
28. M. Chase, Multi-authority attribute based encryption, in *Theory of Cryptography*, Berlin, Heidelberg, (2007), 515–534. https://doi.org/10.1007/978-3-540-70936-7_28
29. S. J. De, S. Ruj, Decentralized access control on data in the cloud with fast encryption and outsourced decryption, in *2015 IEEE Global Communications Conference (GLOBECOM)*, (2015), 1–6. https://doi.org/10.1109/GLOCOM.2015.7417639
30. M. Xiao, Q. Huang, Y. Miao, S. Li, W. Susilo, Blockchain based multi-authority fine-grained access control system with flexible revocation, *IEEE Trans. Serv. Comput.*, **15** (2021), 3143–3155. https://doi.org/10.1109/TSC.2021.3086023
31. K. Sethi, A. Pradhan, P. Bera, PMTER-ABE: a practical multi-authority CP-ABE with traceability, revocation and outsourcing decryption for secure access control in cloud systems, *Cluster Comput.*, **24** (2021), 1525–1550. https://doi.org/10.1007/s10586-020-03202-2
32. P. Datta, I. Komargodski, B. Waters, Decentralized multi-authority ABE for NC 1 from BDH, *J. Cryptology*, **36** (2023), 6. https://doi.org/10.1007/s00145-023-09445-7
33. S. Hohenberger, B. Waters, Online/offline attribute-based encryption, in *Public-Key Cryptography – PKC 2014*, Buenos Aires, Argentina, (2014), 293–310. https://doi.org/10.1007/978-3-642-54631-0_17
34. A. Mughal, A. Joseph, Blockchain for cloud storage security: a review, in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, (2020), 1163–1169. https://doi.org/10.1109/ICICCS48265.2020.9120930