Secure and Flexible Authorized Data Sharing for Smart Grid

Yawen Feng
Xihua University
Shengke Zeng (zengsk@mail.xhu.edu.cn)
Xihua University

Research Article

Keywords: Privacy-preserving Smart Grid, Fine-grained Access Control, Searchable Encryption, Data Sharing

Posted Date: November 15th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3575150/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Peer-to-Peer Networking and Applications on February 13th, 2024. See the published version at https://doi.org/10.1007/s12083-024-01649-3.

Secure and Flexible Authorized Data Sharing for Smart Grid

Yawen Feng and Shengke Zeng

School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China.

*Corresponding author(s). E-mail(s): zengsk@mail.xhu.edu.cn; Contributing authors: feisongan@stu.xhu.edu.cn;

Abstract

Data in smart grid requires security and privacy. Fine-grained access control provides possibility for various electricity companies and organizations to access owner data securely and flexibly for transdiscipline billing. Moreover, revocation should also be considered for key leakage. This work presents a secure authorized data sharing for smart grid scenario. The data access for electricity companies in this scheme is fine-grained and revocation against server-user collusion is achieved. The security analysis and experiment results show that our solution is privacy-aware and practical for smart grid.

Keywords: Privacy-preserving Smart Grid, Fine-grained Access Control, Searchable Encryption, Data Sharing

1 Introduction

Compared with the traditional grid, the smart grid can monitor data in real time and analyze data for prediction. However, as the scale of the smart grid expands, the limited resources of electricity companies are insufficient to support the storage and computation of massive grid data [1, 2]. Electricity companies outsource smart grid data to third-party cloud storage platforms to reduce the burden of local storage and computation. Grid data that is out of physical control would leak customers' privacy, therefore it needs to be encrypted before outsourcing to the cloud server. Meanwhile, the electricity company employees in different positions have different features, they should achieve fine-grained authorization to access customers' electricity data. As shown in the Figure 1, employees at the basic level can access the electricity data of the customers when the customers are checking out the billing service, and engineers of the electricity company can analyze and forecast the data based on the electricity data of the customers. However, querying and accessing data in the form of ciphertext brings new challenges in terms of efficiency and flexibility. Fortunately, fine-grained searchable encryption can effectively solve the problem.

Moreover, the authority of an electricity company employee is not set forever. When an employee leaves the company or the key is lost, the authority of the employee needs to be revoked [3]. If the electricity employees have the same attributes as the revoked authority, then these employees update their attribute key while the server updates the stored ciphertext, so as to revoke the employee's authority. However, the server may be untrustworthy. To obtain information about a customer's data, the server may collude with a malicious revoked-access electricity company employee. The server retains the not-updated ciphertext, while the malicious employee can use the trapdoor generated by the not-updated attribute key to request data from the server. It damages the customer seriously.

Therefore, we focus on revocable attribute-based searchable encryption against server-client collusion to realize privacy-preserving data sharing in smart grid.

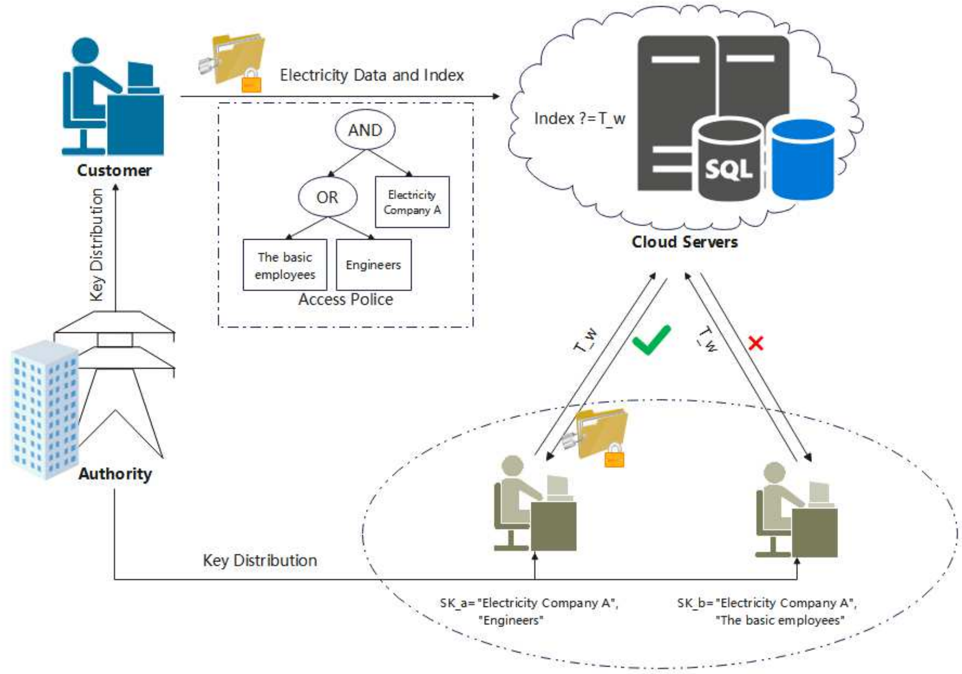


Fig. 1 Flexible Authorized Data Sharing for Smart Grid

1.1 Related Work

Security and data privacy of the smart grid have gained the attention of scholars. Zhao et al.[4] proposed a blockchain-based privacy-protecting billing (BPB) framework to protect customers' private information. The scheme encrypts electricity data before transmitting and storing it. The integrity of the encrypted electricity data can be verified when the user applies for an electricity bill. Mehta et al.[5] proposed a lightweight authentication and key agreement framework [6] based on the protection of electricity data. Egide et al.[7] proposed a heterogeneous authentication protocol for smart grid (HAP-SG). It realizes encrypted communication between communicating entities in different encrypted environments and protects the privacy of user data. The above scheme realizes the security of customer electricity data using authentication. However, it is not used for fine-grained authorization access for the electricity company employees. Attribute-based searchable encryption is essential to solve this problem.

Searchable encryption was first proposed by Song et al.[8] to realize search on encrypted data. Then Boneh et al.[9] extended the range of searchable encryption techniques by proposing a public key searchable encryption scheme. Attribute-based searchable encryption provides fine-grained access control [10, 11] for the data user in searchable encryption. Based on it, Nabeil et al.[12] proposed a kind of online/offline approach to reduce the computation burden for data owner in smart grid. Zhang et al.[13] proposed attribute-based keyword search encryption for power data protection. Only authorized users can retrieve power grid data, while access policy hiding is implemented to protect user privacy. However, they do not take into consideration the revocation of malicious data user authority.

To solve this problem, Zhang et al.[14] proposed a secure revocable data-sharing framework for IoT. The scheme realizes secure data sharing while applying the KUNode algorithm [15] to achieve attribute revocation for malicious users and protect the data owner's privacy. Li et al.[16] proposed the subset coverage attribute revocation technique to solve the grid data privacy leakage problem during data privileges revocation. Yang et al.[17] realized the tracking of data sources and betrayers. The privacy of data owners and users is protected with noninteractive zero-knowledge proofs. Moreover, an update operation is executed on the encrypted file after revealing the malicious user. Ge et al.[18] proposed a direct revocation attribute-based proxy re-encryption scheme. Wang et al.[19] proposed an attribute-based encrypted search for multi-owner and multi-user distributed systems (AESM2). Achieve fine-grained searchable encryption in multi-user scenarios while attribute revocation for malicious data users. In the same year, Niu et al.[20] proposed an attribute-based searchable encryption scheme for lightweight device edge computing. Achieve efficient data access and attribute revocation. The implementation of the scheme [14–20] attribute revocation all depend on the credibility of the server. Some scholars have proposed puncture encryption techniques. Wei et al.[21] and Ghopur et al.[22] implemented user attribute revocation by using punctured encryption. The revocation user is trusted in this scheme. In the case of malicious users may not consciously execute the puncture encryption operation. With the above analysis, the untrustworthy cloud server and the revoked user may collude to obtain the data of the data owner. This harms seriously the benefits of the data owner. Nazatul et al.[23] introduced a third-party attribute verification authority (VA) to verify the user attribute authenticity in the outsourced encrypted data-authorised keyword search scheme. The scheme needs to require the attribute verification authority to be honest. Yu et al.[24] proposed an efficient revocable and searchable MA-ABE scheme with blockchain assistance for C-IoT, which achieves trustworthy revocation through blockchain and smart contracts. However, the leakage of update keys during the execution of attribute revocation can also lead to data security issues. Finally, there is an urgent need for the VA to ensure the authenticity of user attributes in the keyword search scheme.

1.2 Contributions

In this paper, we propose a secure and flexible authorized data sharing scheme for smart grid. The searching trapdoor is publicly verified, so as to solve the problem of untrustworthy cloud servers and malicious electricity company employees collusion. Our contributions are shown as follows:

- Secure and flexible access to electricity data. Enables fine-grained access to encrypted electricity data through attribute-based searchable encryption. It can update the ciphertext in time when the attributes of the electricity company's employees are revoked. It also verifies the validity of the search trapdoor to prevent malicious collusion.
- Update keys public transmission. Update keys for the electricity company employees with unrevoked attributes are publicly available. Only employees with unrevoked privileges can use the update key to generate the latest version of the key. An employee with revoked privileges who gets the update key will not be able to compute a valid search trapdoor.
- Low-communication ciphertext update. The server can update the file cipher and index cipher using public parameters. No extra interaction with the server is required from the client or the authorized authority.

1.3 Organization

The rest of this study is organized as follows. Section 2 presents some preliminary cryptographic background. Section 3 describes the system architecture and definitions. Section 4 proposes our scheme and analyzes the security. Section 5 gives the performance analysis. Finally, Section 6 summarizes the conclusions of this paper.

2 Preliminaries

2.1 Bilinear Mapping and Complexity Assumption

Let G and G_T be multiplicative cyclic groups of order prime q, where g is a generating element of G. Define bilinear pairs $\hat{e} : \mathrm{G} \times \mathrm{G} \rightarrow \mathrm{G}_T$ satisfying the following characteristics [25, 26]:

- Bilinear: For any $a, b \in Z_q^*$ and $g \in \mathrm{G}$, there is $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.
- Non degeneracy: $\hat{e}(g, g) \neq 1$, where $g \in \mathrm{G}$.
- Computability: There is a feasible algorithm to calculate $\hat{e}(g, g)$ for any $g \in \mathrm{G}$.

Bilinear Decisional Diffie-Hellman (*BDDH*) Assumption is over bilinear mapping. The security of our scheme is based on *BDDH* assumption.

Definition 1 (*BDDH* Assumption). *For given* $\left(g, g^a, g^b, g^c, g^d\right) \in \mathrm{G}$, *where* $a, b, c, d \in Z_q^*$, *it is difficult to decide* $\hat{e}(g,g)^{abc} \overset{?}{=} g^d$.

2.2 Access Structure

Let $A = \{A_1, A_2, \cdots A_n\}$ denote attributes universe [27]. Then $\mathcal{A} \subseteq 2^{\{A_1, A_2, \cdots A_n\}}$ is a non-empty subset of $\{A_1, A_2, \cdots A_n\}$. And the set $\mathcal{A}$ is monotonic for any set of attributes B, C: if $B \subseteq \mathcal{A}$ and $B \subseteq C$, then $C \subseteq \mathcal{A}$. The sets in $\mathcal{A}$ are known as authorized sets, otherwise, it is known as non-authorized sets.

2.3 Linear Secret-Sharing Schemes

There exists a $l \times n$ secret generation matrix M, and the row i of the matrix M is denoted as M_i, where ρ_i denotes the mapping from M_i to attribute A by ρ calculation [28]. We define the access structure as $\mathcal{L} = (M, \rho)$. Select randomly $s, \nu_2, ..., \nu_n \in Z_q$. Define the column vector $\vec{\nu}$, where the first real number in the $\vec{\nu}$ is used as the secret value. Then $\lambda_i = M_i \cdot \vec{\nu}$ is the l vector containing some of the secret values s calculated according to M. Let U be an attribute set of the user where $U \in \mathcal{A}$, and define $I = \{i : \rho(i) \in U\}$ $(i \subseteq 1, 2, ..., l)$. There exists a factor $\{c_i \mid i \in I\}$ so that $\sum_{i \in I} c_i M_i = (1, 0, 0, ..., 0)$. Thus the calculation $\sum_{i \in I} c_i \lambda_i = s$ is obtained.

2.4 Time Binary Tree

We follow the strategy of scheme [27] to implement the user attribute revocation. Embed the time period in the ciphertext, and the depth d of the binary tree structure BT_t is constructed to manage the time period, where N_t leaf nodes denote $\left\{0, 1, ..., 2^d - 1\right\}$ of discrete time periods. The root node of the time binary tree is denoted as R_t, the leaf nodes as σ_t and the non-leaf nodes as σ, where the left and right children of the non-leaf nodes are denoted as σ_l and σ_r respectively. Let b_{σ_t} denote the path from the root node to the leaf nodes, where traversing σ_l is written as 0 and traversing σ_r is written as 1. When this time period is t, define $N_t = \{\sigma_r \mid \sigma \in Path(\sigma_t) \wedge \sigma_r \notin Path(\sigma_t)\} \cup \{\sigma_t\}$. If $\hat{t} > t$, this has a node $\sigma_{\hat{t}} \in N_{\hat{t}}$ that makes string b_{σ_t} a prefix of string $b_{\sigma_{\hat{t}}}$ for each node $\sigma_t \in N_t$.

3 System Architecture and Definitions

In this section, We introduce a framework for secure and flexible authorized data sharing for smart grid. We also describe the three aspects of the system architecture, its syntax and the security model.

3.1 System Architecture

The following 6 entities are included in the system: Data Owner, Data Users, Public Users, Cloud Server, Authorisation Centre and Attribute Authorities.

- **Data Owner (*DO*).** The data owner is a customer in the smart grid who has electricity data. The customer's electricity data needs to be designated with an access policy based on the attributes of the electricity company employees before it is uploaded to the cloud server. And the customer encrypts the files and keywords to generate the cipher components. Then DO uploads the cipher component to the server.
- **Data users (*DU*).** The data users are the electricity company employees, and employees in different functions have different authorities. Each electricity company employee is given an identity and sends his attributes to the attribute authorities to get the key. The attribute authority provides the non-revoked employee with an update key to help employees generate the latest valid attribute key after the attribute is revoked.
- **Public Users (*PU*).** Public users are public parties. They can be any user in the system. When the customer uploads the trapdoor to the server, the public users verify the authenticity of the trapdoor permissions.
- **Cloud Server (*CS*).** The main responsibility of the cloud server is to store encrypted electricity data uploaded by customers. Execute the ciphertext update operation when the authority of the electricity company employees changes. It helps the employees to run a keyword search operation when the electric company employees access stored ciphertext.
- **Authorisation Centre (*AC*).** This is primarily responsible for initializing. The AC generates the master key and the system public key. Then it generates partial keys for the electricity company employees.
- **Attribute Authorities (*AAs*).** Each attribute authority manages its attributes independently. At system initialization, generate its public-private key pairs. Generate attribute keys for electric company employees when they request data. In case of attribute revocation, update keys are provided for non-revoked employees. And AAs publish the authorization information to help the public party verify the employee's trapdoor.

3.2 Syntax

This paper mainly solves the problem of collusion between semi-trusted servers and revoked users in the existing smart grid scheme for attribute revocation. Untrustworthy servers may keep copies of ciphertexts, and malicious employees use trapped request data generated by attribute keys with revoked privileges, then their collusion seriously harms the interests of customers. This paper adopts the method of publicly verifying the validity of trapped doors for solid secure and flexible authorization of smart grid data sharing, in which the time function in the scheme is publicly available. The public users can use the time function and the authorization authority's publicly available authorization information to verify the validity of the trapdoor for electricity company employees. This ensures the integrity of data users and limits the behavior of servers. Under public scrutiny, only a valid trapdoor can cause the server to return the ciphertext requested by the electricity company employee. This prevents the server from colluding with malicious employees. Besides, due to the keyword

Table 1 Notation Definitions

Notions	Descriptions
$\mathcal{L}$	Access structure
U_a, U_A	Attributes universe and authorisations universe
$\Phi()$	Mapping of functions u to attribute
$\Psi()$	$\Psi(x) = \Phi(\rho(x))$
N_u	Electricity company employees
ζ	Nodes of a user binary tree
N_b	Minimum node set
RL_θ	Revocation list
N_t	Discrete time period
σ	Nodes of a time binary tree
b_{σ_t}	The binary string of the σ
SK_{ID,U_θ}	Secret key of DU
$UK_{\theta,t}$	Update key of DU
F	File
CT	Ciphertext components
w	Keyword
I_w	Index for keyword w
TK^t_{ID,U_θ}	Trapdoor

index ciphertext and the trapdoor privacy, the cloud server can only execute keyword-matching operations and can't get the plaintext of any keyword. The system model of the scheme is shown in Figure 2. We describe the notations that will be used in the scheme construction in Table 1.

(1) **GlobalSetup**$(\kappa) \rightarrow (PP, MSK)$: This algorithm is executed by AC. Given a security parameter κ, the AC calculates the master key MSK and the system public parameters PP based on κ.

(2) **AuthSetup**$(PP, T, N_u) \rightarrow (PK_\theta, SK_\theta, RL_\theta, ST_\theta)$: AAs execute the algorithm. Input the system public parameters PP, the total time periods T and the number of the electricity company employees N_u, output the public-private key pair (PK_θ, SK_θ) of the AAs, an empty revocation list RL_θ and status information ST_θ to manage the status information of attribute revocation employees and their identifiers, where $\theta \in \mathcal{A}$.

Remark 1. *PP and PK_θ are both contained to be denoted as PK. MSK and SK_θ are both contained to be denoted as SK.*

(2) **KeyGen**$(PP) \rightarrow (SK_{DO}, PK_{DO})$: The algorithm is executed by the customer. The customer enters system public parameters PP and outputs its own public-private key pairs (SK_{DO}, PK_{DO}).

(3) **Encrypt**$(PK, \mathcal{L}, F, w, t) \rightarrow (CT, I_w)$: The customer executes the algorithm. Input the public key PK, the access policy $\mathcal{L}$, the file F, the keyword w and the time period t, then output the file cipher CT and keyword index cipher I_w.

(4) **SKeyGen**$(PK, SK, ID, U_\theta, ST_\theta) \rightarrow (D, SK_{ID,U_\theta})$: Phase I, This algorithm is executed by AC. Enter the master key MSK and the system public parameters PP, then the AC sends the partial key D to the electricity company employee. Phase II, This algorithm is executed by AAs. Enter the system public parameters PP,

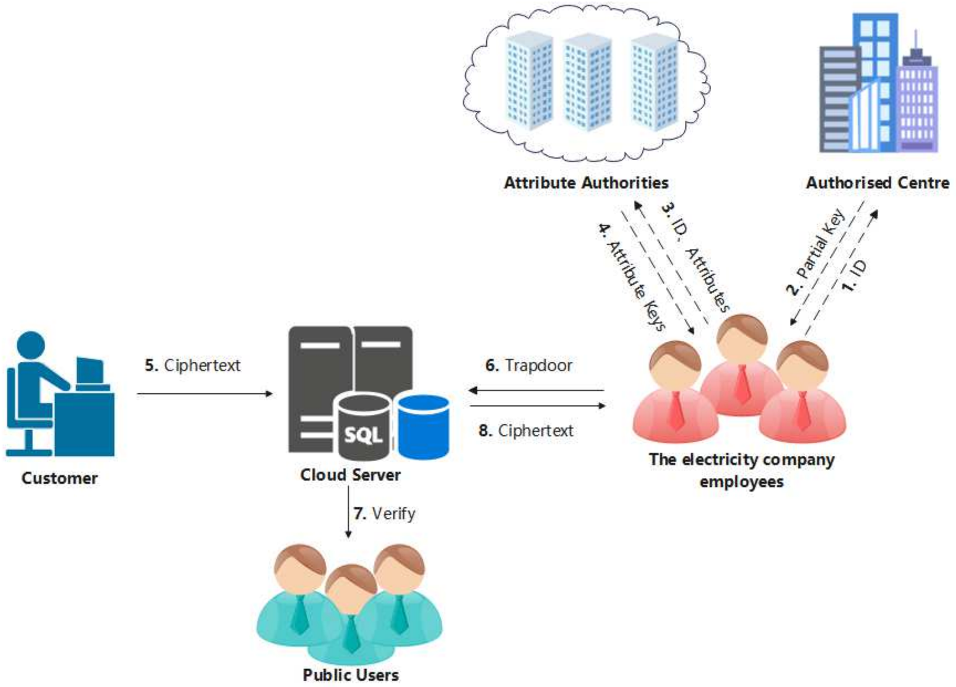


Fig. 2 System Model

the public and private key pair (PK_θ, SK_θ) of the attribute authority, the global identifier ID and attribute set U_θ of the employee and the status information ST_θ; then the AAs update the status information text ST_θ of the employee's identifier and generate the attribute key SK_{ID,U_θ}.

(5) **UKeyGen**$(PK, SK_\theta, RL_\theta, ST_\theta, t) \rightarrow (UK_{\theta,t}, W_\theta(t), P_{ID,t})$: AAs execute the algorithm. Input the public key PK, the private key SK_θ of the attribute authority, the attribute revocation list RL_θ, the status information ST_θ and the period t; then the AAs publish the authorization information $P_{ID,t}$.

(6) **Trapdoor**$(PK, w, SK_{ID,U_\theta}, UK_{\theta,t}) \rightarrow \left(TK^t_{ID,U_\theta}\right)$: This algorithm is executed by the electricity company employees. Take the public key PK, the keyword w, the attribute key SK_{ID,U_θ} and the update key $UK_{\theta,t}$ as input. Finally, the electricity company employee generates the trapdoor TK^t_{ID,U_θ}.

(7) **Verify**$(PK, TK^t_{ID,U_\theta}, CT, P_{ID,t})$: This algorithm is executed by TA. Input the public key PK, the authorization information P of server, and the trapdoor TK^t_{ID,U_θ}. The TA verifies that the trapdoor is the correct authorization key generation. When the verification passes, the CS executes a search operation.

(8) **Search**$\left(PK, CT, I_w, TK^t_{ID,U_\theta}\right) \rightarrow (C, Q)$: CS executes the algorithm. Enter the public key PK, the trapdoor TK^t_{ID,U_θ}, the cipher component (CT, I_w) and calculation result Q. The CS exexutes a keyword search operation and returns the

searched ciphertext C to the electricity company employee when the keyword match is passed. Then the CS sends (C, Q) to the employee.

(9) **Decrypt**$(PP, C, Q, TK^{t}_{ID,U_\theta}) \rightarrow (F)$: This algorithm is executed by the electricity company employee. Take the system parameters PP, the trapdoor TK^{t}_{ID,U_θ} and the cipher component $CT^{'}$ as input. Then the employee performs the decryption operation to obtain the file F.

(10) **CTUpdate**$(PK, CT, t^{'}) \rightarrow (CT_{t'})$: This algorithm is executed by CS. Input the public key PK, the ciphertext component CT and the time period $t^{'}$, then output the updated ciphertext component $CT_{t'}$.

(11) **Revoke**$(ID, RL_\theta, ST_\theta, t) \rightarrow (RL^{t}_{\theta})$: This algorithm is executed by AAs. Input the employee's identification ID, attribute revocation list RL_θ, status information ST_θ and the time period t, then output the attribute revocation list RL^{t}_{θ} after updating.

3.3 Security Model

3.3.1 Keyword Indistinguishability

We describe the indistinguishability of keyword w as a game between the attacker $\mathbb{A}$ and the challenger $\mathbb{C}$, where the attacker's advantage ε in winning is negligible.

- **Setup:** The $\mathbb{A}$ sends the security parameters to the $\mathbb{C}$, who initializes the system and generates the system public parameters PK and the main key MSK. Then $\mathbb{C}$ publicized PK. The $\mathbb{A}$ selects an access policy $\mathcal{L}^{'}$ to send to the $\mathbb{C}$ as a challenge.
- **Phase1:** The attacker sends a keyword to the challenger and requests the trapdoor corresponding to the keyword. The attacker sends multiple ciphertext requests to the $\mathbb{C}$.

  – Trapdoor Oracle($\mathcal{O}_T$): The challenger maintains a list L_{tw} of recorded keyword trapdoor pairs. The $\mathbb{C}$ calculates the keyword trapdoor T_w based on **Trapdoor**$(PK, w, SK_{ID,U_\theta}, UK_{\theta,t})$, where SK_u is generated by **SKeyGen**$(PK, SK, ID, U_\theta, ST_\theta)$. Then the $\mathbb{C}$ records the computed keyword trapdoor pairs (w, T_w) into L_{tw}.

- **Challenge:** The $\mathbb{A}$ selects two keywords (w_0^*, w_1^*) of equal length and requests the trapdoor. The $\mathbb{C}$ executes the trapdoor generation algorithm and generates trapdoor $T_{w_b^*}$ corresponding to the keywords, which belong to $b \in \{0, 1\}$. And (w_0^*, w_1^*) are not queried before.
- **Phase2:** Consistent with Phase1.
- **Guess:** The $\mathbb{A}$ outputs the guessed answer $b^{'}$. Here is the probability of $\mathbb{A}$ winning a keyword indistinguishability game.

$$Adv\left(1^{\kappa}\right) = \left|Pr\left[b^{'} = b\right] - \tfrac{1}{2}\right|$$

4 Detailed Construction of the Scheme

4.1 Algorithm Description

The specific steps of the programme are as follows:

GlobalSetup(κ): The AC is primarily responsible for initialising the system. Given a security parameter κ, two multiplicative cyclic groups G and G_T of order q are generated by the authorisation centre, where $g \in G$. Define bilinear pairs $\hat{e} : G \times G \to G_T$.

1. Choose a secure hash function $H : \{0,1\}^* \to G$.
2. Select randomly number components $a, b, c \in Z_q$ and calculate g^a, g^b, g^c,$\hat{e}(g,g)$, $\hat{e}(g,g)^{ac}$.

Expose the main key $MSK = (a, b, c)$ and the system public parameters $PP = (G, G_T, g, q, g^a, g^b, g^c, \hat{e}, \hat{e}(g,g), \hat{e}(g,g)^{ac}, H)$.

AuthSetup(PP, T, N_u): For each authority $\theta \in \mathcal{A}$, U_θ represents the attribute managed by the authority θ. The AAs are initialised to generate an empty revocation list RL_θ, a temporal binary tree BT_t of depth d with $T = 2^d$ time periods and a user binary tree BT_θ with leaf nodes N_u.

1. Select randomly number components $\alpha_\theta, \beta_\theta \in Z_q$ and group element components $f_{\theta,0}, f_{\theta,1}, ..., f_{\theta,d} \in G$.
2. Represent the state information as ST_θ and N_θ represent all nodes in the user binary tree BT_θ as and pick a random number $r_\zeta \in Z_q$ for each node N_θ.
3. Define a function $W_\theta(t) = f_{\theta,0} \prod_{j=1}^{d} f_{\theta,j}^{t[j]}$, where $t\,[j]$ represents the jth bit in the binary string of time period t.

Finally, set the public key to $PK_\theta = (\hat{e}(g,g)^{\alpha_\theta}, g^{\beta_\theta}, f_{\theta,0}, f_{\theta,1}, ..., f_{\theta,d})$ and the private key to $SK_\theta = (\alpha_\theta, \beta_\theta, \{r_\zeta\}_{\zeta \in N_\theta})$.

KeyGen (PP): The customer executes this algorithm to generate his own public-private key pairs, who selects randomly an element $x \in Z_q^*$ as private key and computes the corresponding public key g^x. Let $(SK_{DO} = x, PK_{DO} = g^x)$.

Encrypt$(PK, \mathcal{L}, F, w, t)$: The customer designates an access policy $\mathcal{L} = (M, \rho)$ based on the attributes of the electricity company employees, where M is a $l \times n$ secret generation matrix. Then the customer encrypts the electricity data F and the keyword w.

1. Selected randomly $\vec{x} = (s, x_2, ..., x_n)$ and $\vec{y} = (0, y_2, ..., y_n) \in Z_q^n$, where the first real number in the $\vec{x}$ is used as the secret value s, then $\lambda_i = M_i \cdot \vec{x}$ is the l vector containing some of the secret values s calculated according to M. And $\pi_i = M_i \cdot \vec{y}$.
2. The customer uses the access policy to encrypt F to generate the ciphertext components at a time period t. Select randomly number components $z_i \in Z_q (i \in [l])$ and calculate

$$\begin{aligned} &C = F \cdot \hat{e}(g,g)^{acs}, \\ &C_{i,1} = \hat{e}(g,g)^{ac\lambda_i} \hat{e}(g,g)^{\alpha_{\rho(i)} z_i}, \\ &C_{i,2} = g^{-z_i}, C_{i,3} = g^{\beta_{\rho(i)} z_i} g^{\pi_i}, \\ &C_{i,4} = g^{H(\Psi(i)) z_i} g^{\pi_i} \end{aligned}$$

For each node $\gamma \in N_t$.

$$C_{i,\gamma} = \left(C_{i,\gamma,0}, C_{i,\gamma,|b_\gamma|+1}, ..., C_{i,\gamma,d}\right),$$

where

$$C_{i,\gamma,0} = \left(f_{\rho(i),0} \prod_{j=1}^{|b_\gamma|} f_{\rho(i),j}^{b_\gamma|j|} \right)^{z_i},$$

and

$$C_{i,\gamma,k} = f_{\rho(i),k}^{z_i} \text{ for } k \in [\mid b_\gamma \mid +1, d].$$

3. The customer extracts the keywords from the electricity data F. Then DO selects randomly an number $r \in Z_q^*$ and calculates the keyword indexed ciphertext $I_w = g^{a(r+s)} g^{brxH(w)}, I_1 = g^{cr}$.
4. Return the ciphertext CT.

$$\left\{ C, \left\{ C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, \{C_{i,\gamma}\}_{\gamma \in BT_t} \right\}_{i \in [l]}, I_w, I_1, \mathcal{L}, t \right\}$$

SKeyGen$(PK, SK_\theta, ID, U_\theta, ST_\theta)$: The electricity company employee calculates $g^{xH(ID)}$ by the public key g^x of the customer, and he sends $g^{xH(ID)}$ to the authorisation centre. Then the DU randomly picks a element $\mu \in Z_q$ and computes $g^{\frac{H(ID)}{\mu}}$. His global identity ID, $g^{\frac{H(ID)}{\mu}}$ and attribute set $S_\theta \subseteq U_\theta$ are sent to the attribute authorities.

1. The AC uses the PP and MSK to generate a private key $D = g^{bxH(ID)}$ for the employee.
2. The AAs choose an unassigned leaf node τ_t to store the employee's ID. For each node $\zeta \in Path(\tau)$, select randomly a number $\delta_u \in Z_q$ for each attribute $u \in S_\theta$ and select randomly a number $\delta_{ID} \in Z_q$ for each employee's ID. Then generate the attribute key component $SK_\zeta = (D, \left\{ K_{\zeta,\theta,u,ID}, K'_{\zeta,\theta,u,ID}, K_{\zeta,u} \right\}_{u \in S_\theta})$ by using the private key SK_θ of AAs. Meanwhile, AAs update state information ST_θ.

$$\begin{aligned} K_{\zeta,\theta,u,ID} &= g^{\alpha_\theta - r_\zeta} H(ID)^{\frac{\beta_\theta}{\mu}} g^{H(u)\delta_u}, \\ K'_{\zeta,\theta,u,ID} &= g^{\delta_u}, \\ K_{\zeta,u,ID} &= g^{cH(u)\frac{r_\zeta}{H(ID,\delta_{ID})}} \end{aligned}$$

Output the secret key $SK_{ID,S_\theta} = \{SK_\zeta\}_{\zeta \in Path(\tau)}$.

The part of the key D generated by the electricity company employee interacting with the AC. D is a component of the keyword trapdoor. Constructing keyword trapdoors based on the $BDDH$ problem satisfies privacy.

UKeyGen$(PK, SK_\theta, RL_\theta, ST_\theta, t)$: The electricity company employee with an unrevoked attribute generates an update key based on the newly published revocation list RL_θ and status information ST_θ. The steps are as follows

1. The set of nodes M_θ is obtained by running the **KUNode** algorithm, where $M_\theta \subseteq N_\theta$.
2. Select randomly a number $\varepsilon_t \in Z_q$ for each node $\zeta \in M_\theta$ and calculate $UK_{\zeta,\theta} = (U_{\zeta,\theta}, U'_{\zeta,\theta}, R_{\zeta,\theta}) = (g^{r_\zeta} \cdot W_\theta(t)^{\varepsilon_t}, g^{\varepsilon_t}, \frac{\varepsilon_t}{r_\zeta})$ by using the PK and SK_θ.

3. AAs publish the authorization information $P_{ID,t} = \frac{\varepsilon_t}{H(ID,\delta_{ID})}$ to help the public to verify the electricity company employee's trapdoor.

The AAs send the secret key $UK_{\zeta,t} = \{UK_{\zeta,\theta}\}_{\zeta \in M_\theta}$ to employees through the secure channel. Only key components that contain the same node r_ζ can use the update key.

Trapdoor$(PK, w, SK_{ID,U_\theta}, UK_{\theta,t})$: When the employee wants to access the ciphertext stored on the server, the DU uses the random number μ to calculate the following trapdoor.

1. Then DU computes $D^{'} = (D)^{\frac{\mu H(w)}{H(ID)}} = g^{bxH(w)\mu}$ using his own known keyword w and partial key D.
2. Find out $\zeta \in M_\theta \cap Path(\tau)$ from SK_{ID,U_θ} and $UK_{\theta,t}$. Choose a random exponent $\varepsilon_t^{'} \in Z_q$ and calculate $T = (T_{\theta,u,ID}, T^{'}_{\theta,u,ID}, T^{'}_{\theta,t}, T_{\zeta,u}, T_\mu)$.

$$\begin{aligned}
&T_w = g^{a\mu} \cdot D^{'} \\
&T_{\theta,u,ID} = (K_{\zeta,\theta,u,ID} \cdot U_{\zeta,\theta} \cdot W_\theta(t)^{\varepsilon_t^{'}})^\mu, \\
&T^{'}_{\theta,u,ID} = (K^{'}_{\zeta,\theta,u,ID})^\mu, \\
&T_{\theta,t} = (U^{'}_{\zeta,\theta} \cdot g^{\varepsilon_t^{'}})^\mu, \\
&T_{\zeta,u,ID} = (K_{\zeta,u,ID})^{\mu R_{\zeta,\theta}}, T_\mu = g^{c\mu}
\end{aligned}$$

The electricity company employee submits the search trapdoor $T^t_{ID,S_\theta} = (T_w, T_{\theta,t}, T_\mu, \left\{T_{\theta,u,ID}, T^{'}_{\theta,u,ID}, T_{\zeta,u}\right\}_{u \in S_\theta})$ to the cloud server, where $(T_w, T_{\theta,t}, \left\{T_{\theta,u,ID}, T^{'}_{\theta,u,ID}\right\}_{u \in S_\theta})$ is used to execute the keyword search operation and $(\{T_{\zeta,u}\}_{u \in S_\theta}, T_\mu)$ is used to verify that the DU is correctly authorized.

Verify$(PK, TK^t_{ID,U_\theta}, CT, P_{ID,t}, W_\theta(t))$: At this phase, the public can verify that the server has executed the re-encryption operation based on the time function $W_\theta(t)$. Then it verifies that the electricity company employee is the latest authority based on the authorization information $P_{ID,t}$.

1. Input the time function $W_\theta(t)$ and the ciphertext component CT to judge whether the following equation holds. If the following equation passes continue to execute 2), otherwise interrupt. If the time period t is determined then time function $W_\theta(t)$ must be determined, and judge from $W_\theta(t)$ whether the ciphertext CT matches the following equation.

$$\hat{e}(W_\theta(t), C_{i,2}) \cdot \hat{e}(C_{i,\gamma,0}, g) = 1$$

2. From the above equation we can judge that the ciphertext component CT is the correct ciphertext for t. For each $i \in I$, compute

$$\prod_{i \in I} (\hat{e}(C_{i,4}^{P_{ID,t}}, T_\mu))^{c_i} \cdot \hat{e}(T_{\zeta,u}, C_{i,2}) = 1$$

The PU uses the CT and the authorization information $P = \frac{\varepsilon_t}{H(ID,\delta_{ID})}$ to verify the authority of the electricity company employee. The employee who has the ε_t must hold the latest update key to generate the trapdoor. If the above equation passes continue to execute 3), otherwise interrupt.

Search $\left(PK, CT, I_w, TK^t_{ID,U_\theta}\right)$: After the server receives the trapdoor sent by the electricity company employee. There is an interactive process between the cloud server and the employee. Then the CS executes a keyword search operation to match the keyword indexed ciphertext I_w with the trapdoor TK^t_{ID,U_θ}.

1. The server sends $C_{i,1}$ to the employee, who returns the calculated result $C^{\mu}_{i,1} = (C_{i,1})^{\mu}$ to the server.
2. For each $i \in I$, calculate Q_i and $Q = \prod_{i\in I} Q_i^{c_i}$.

$$\begin{aligned} Q_i = C^{\mu}_{i,1} \cdot \hat{e}(T_{\theta,u,ID}, C_{i,2}) \cdot \hat{e}(H(ID), C_{i,3}) \\ \cdot \hat{e}(T^{'}_{\theta,u,ID}, C_{i,4}) \cdot \hat{e}(T_{\theta,t}, C_{i,\gamma,0}) \end{aligned}$$

3. Calculate the following equation using the calculation result Q. Check if the electricity company employee matches the access policy. And whether the keywords in the ciphertext match or not.

$$\hat{e}(T_w, I_1) \cdot Q = \hat{e}(I_w, T_\mu)$$

When the search trapdoor TK^t_{ID,U_θ} and the keyword w satisfy the above equation, the server returns the cipher (C, Q) corresponding with the keyword index cipher to the employee.

Decrypt$(PP, C, Q, TK^t_{ID,U_\theta})$: When the electricity company employee receives the ciphertext C returned by the server. The employee calculates the decryption key $k = Q^{\frac{1}{\mu}}$ by random number μ. Then DU calculates $\frac{C}{k} = \frac{F\cdot\hat{e}(g,g)^{acs}}{\hat{e}(g,g)^{acs}}$. The ciphertext C is decrypted to obtain the plaintext F.

CTUpdate$(PK, CT, t^{'})$: When the electricity company employee's attributes are revoked. To protect customer's privacy, the server needs to execute a re-encryption operation to update the stored ciphertext at a new time period $t^{'}$.

1. Select randomly $\vec{x^{'}} = (s^{'}, x^{'}_2, ..., x^{'}_n)$ and $\vec{y^{'}} = (0, y^{'}_2, ..., y^{'}_n) \in Z^n_q$, where the first real number in the $\vec{x^{'}}$ is used as the secret value $s^{'}$, then compute $\lambda^{'}_i = M_i \cdot \vec{x^{'}}$ and $\pi^{'}_i = M_i \cdot \vec{y^{'}}$.
2. Select randomly number components $z^{'}_i \in Z_q$ for each $i \in [l]$ and calculate

$$\begin{aligned} &C^{'} = C \cdot (\hat{e}(g,g)^{ac})^{s^{'}}, \\ &I^{'}_w = I_w \cdot (g^a)^{s^{'}}, \\ &C^{'}_{i,1} = C_{i,1} \cdot \hat{e}(g,g)^{ac\lambda^{'}_i} \hat{e}(g,g)^{\alpha_{\rho(i)} z^{'}_i}, \\ &C^{'}_{i,2} = C_{i,2} \cdot g^{-z^{'}_i}, C^{'}_{i,3} = C_{i,3} \cdot g^{\beta_{\rho(i)} z^{'}_i} g^{\pi^{'}_i}, \\ &C^{'}_{i,4} = C_{i,4} \cdot g^{H(\Psi(i)) z^{'}_i} g^{\pi^{'}_i}. \end{aligned}$$

For each node $\gamma^{'} \in N_{t'}$, calculate

$$C_{i,\gamma'} = \left(C_{i,\gamma',0}, C_{i,\gamma',|b_{\gamma'}|+1}, ..., C_{i,\gamma',d}\right),$$

where

$$C_{i,\gamma',0} = C_{i,\gamma,0} \cdot \prod_{j=|b_\gamma|+1}^{|b_{\gamma'}|} (C_{i,\gamma,j})^{b_{\gamma'}|j|} \cdot \left(f_{\rho(i),0} \prod_{j=1}^{|b_{\gamma'}|} f_{\rho(i),j}^{b_{\gamma'}|j|}\right)^{z_i^{'}},$$

and

$$C_{i,\gamma',k} = C_{i,\gamma,k} \cdot f_{\rho(i),k}^{z_i^{'}} \text{ for } k \in \left[| \ b_{\gamma'} \ | +1, d\right].$$

3. Updated cipher components $CT_{t'}$.

$$\left\{C^{'}, C_1^{'}, \left\{C_{i,1}^{'}, C_{i,2}^{'}, C_{i,3}^{'}, C_{i,4}^{'}, \{C_{i,\gamma'}\}_{\gamma' \in BT_{t'}}\right\}_{i \in [l]}, t^{'}\right\}.$$

Revoke$(ID, RL_\theta, ST_\theta, t)$: When the employee's attributes are revoked at the time period t. Put employee's information in the new revocation list RL_θ^t,and output the status information ST_θ^t.

4.2 Correctness

We show the correctness of the above equations in this section.

Correctness of (1). In the search phase:

$$\begin{aligned}
P_i = {} & C_{i,1}^{\mu} \cdot \hat{e}(T_{\theta,u,ID}, C_{i,2}) \cdot \hat{e}(H(ID), C_{i,3}) \\
& \cdot \hat{e}(T_{\theta,u,ID}^{'}, C_{i,4}) \cdot \hat{e}(T_{\theta,t}, C_{i,\gamma,0}) \\
& = \hat{e}(g,g)^{ac\lambda_i\mu} \hat{e}(g,g)^{\alpha_{\rho(i)} z_i \mu} \\
& \cdot \hat{e}(g^{\mu\alpha_\theta} H(ID)^{\beta_\theta} g^{H(u)\mu\delta_u} W_\theta(t)^{\mu\varepsilon_t + \mu\varepsilon_t^{'}}, g^{-z_i}) \\
& \cdot \hat{e}(H(ID), g^{\beta_{\rho(i)} z_i} g^{\pi_i}) \\
& \cdot \hat{e}(g^{\mu\delta_u}, g^{H(\Psi(i)) z_i} g^{\pi_i}) \\
& \cdot \hat{e}(g^{\mu\varepsilon_t + \mu\varepsilon_t^{'}}, \left(f_{\rho(i),0} \prod_{j=1}^{|b_\gamma|} f_{\rho(i),j}^{b_\gamma|j|}\right)^{z_i}) \\
& = \hat{e}(g,g)^{ac\lambda_i\mu} \cdot \hat{e}(H(ID), g)^{\pi_i} \hat{e}(g,g)^{\mu\delta_u\pi_i}
\end{aligned}$$

and

$$Q = \prod_{i \in I} Q_i^{c_i} = \hat{e}(g,g)^{acs\mu}$$

Then we calculate

$$\begin{aligned}\hat{e}(I_w, T_\mu) &= \hat{e}(g^{a(r+s)} g^{brxH(w)}, g^{c\mu}) \\ &= \hat{e}(g^{a\mu} g^{b\mu xH(w)}, g^{cr}) \cdot \hat{e}(g^{as}, g^{c\mu}) \\ &= \hat{e}(T_w, I_1) \cdot Q\end{aligned}$$

Correctness of (2). In the validation phase: First, we judge whether the ciphertext uploaded by the server is correct or not.

$$\begin{aligned}&\hat{e}(W_\theta(t), C_{i,2}) \cdot \hat{e}(C_{i,\gamma,0}, g) \\ &= \hat{e}(W_\theta(t), g^{-z_i}) \cdot \hat{e}(\left(f_{\rho(i),0} \prod_{j=1}^{|b_\gamma|} f_{\rho(i),j}^{b_\gamma|j|}\right)^{z_i}, g) \\ &= 1\end{aligned}$$

Then judge whether the the electricity company employee's trapdoor is correct or not.

$$\begin{aligned}&\prod_{i\in I} (\hat{e}(C_{i,4}^{P_{ID,t}}, T_\mu))^{c_i} \cdot \hat{e}(T_{\zeta,u}, C_{i,2}) \\ &= \prod_{i\in I} (\hat{e}(g^{H(\Psi(i))z_i} g^{\pi_i})^{\frac{\varepsilon_t}{H(ID,\delta_{ID})}}, g^{c\mu}))^{c_i} \\ &\cdot \hat{e}(g^{cH(\Psi(i))\frac{\mu\varepsilon_t}{H(ID,\delta_u)}}, g^{-z_i}) \\ &= 1\end{aligned}$$

4.3 Security Proof

The scheme's security relies on data confidentiality and keyword indistinguishability. **Theorem 1** (Data Confidentiality)**.** *To solve the collusion between the cloud server and the malicious electricity company employees after the revocation of the attributes in the scheme, we achieve the protection of customer privacy and make the following security analysis.*

Proof. After a malicious electricity company employee revokes an attribute, the authorized organization assists the valid electricity company employee in executing a key update operation to generate a new attribute key to ensure backward security [14]. The update key is calculated according to the KUNode algorithm, and the revoked user cannot execute the key update operation. Thus, only a valid trapdoor can be used to execute the ciphertext search operation. Meanwhile, the server uses the time function to update the stored electricity ciphertext data to realize forward security. And since the re-encryption process calculated by the time function is irreversible, the updated electricity ciphertext data cannot be accessed by the revoked electricity company employee. In addition, in order to prevent the server from keeping a copy of the electricity ciphertext data and the revoked electricity company employee from obtaining the customer's electricity data, the re-encrypted ciphertext of the server and the trapdoor submitted by the electricity company employee are verified publicly. First, the public user uses the time function to verify the electricity ciphertext data, and the time function itself does not contain any customer information. The public user can only get the result of whether the current ciphertext matches the current time function and will not get any plaintext information of the customer. Second, the public user uses the verified ciphertext component and the authorization information disclosed by the authorization center to verify the validity of the trapdoor for the electricity company's employees. Among them, the authorization information is the version key after blinding, which can only verify the trapdoor corresponding to the version key under the current time, and does not disclose any other information.

□

Theorem 2 (Keyword Indistinguishability)**.** *If the BDDH is insolvable, the keywords in the scheme are secure by the following game.*

Proof. If the attacker can break the indistinguishability of the keyword by a non-negligible advantage ε. Then we can simulate a game between the attacker $\mathbb{A}$ and the challenger $\mathbb{C}$ in which the $\mathbb{C}$ can break the *BDDH* problem by the same advantage ε. Technically, $\mathbb{C}$ is given a challenge $\left(g, g^{\alpha}, g^{\beta}, g^{\omega}, g^{d}\right)$ where $\alpha, \beta, \omega, d \in Z_q^*$, its goal is to distinguish $g^{\alpha\beta\omega} = g^{d}$ and g^{d} is random in $\mathcal{G}$.

- **Setup:** The $\mathbb{A}$ sends the security parameters to the $\mathbb{C}$, who initialises the system and generates the system public parameters $PP = (G, G_T, g, q, g^a, g^b, g^c, \hat{e}, \hat{e}(g,g)^{ac})$ and the main key $MSK = (a, b, c)$, then sends PP to the $\mathbb{A}$. The attacker designates an access policy $\mathcal{L}^{'}$ to send to the $\mathbb{C}$ as a challenge.
- **Phase1:** The $\mathbb{A}$ sends a keyword to the $\mathbb{C}$ and requests the trapdoor corresponding to the keyword. The attacker sends multiple ciphertext requests to the $\mathbb{C}$.

  – Trapdoor Oracle($\mathcal{O}_T$): The $\mathbb{C}$ maintains a list L_{tw} of recorded keyword trapdoor pairs. Let $g^{x} = g^{\alpha}$, $g^{b} = g^{\beta}$ and $g^{\mu} = g^{\omega}$, the $\mathbb{C}$ calculates the keyword trapdoor $T_w = g^{a\omega} g^{dH(w)}$ by running **Trapdoor**$(PK, w, SK_{ID,U_\theta}, UK_{\theta,t})$, where $g^{\alpha\beta H(ID)}$ passed through a secure channel is generated by **SKeyGen**$(PK, SK, ID, U_\theta, ST_\theta)$. Then the $\mathbb{C}$ records the computed keyword trapdoor pairs (w, T_w) into L_{tw}.

- **Challenge:** The $\mathbb{A}$ selects two keywords (w_0^*, w_1^*) of equal length and requests the trapdoor. The $\mathbb{C}$ executes the trapdoor generation algorithm and generates trapdoor $T_{w_c^*}$ corresponding to the keywords, which belong to $\tau \in \{0,1\}$. If $\tau = 0$, compute $T_{w_0^*} = g^{a\omega} g^{dH(w_0^*)}$; when $\tau = 1$, compute $T_{w_1^*} = g^{a\omega} g^{\alpha\beta\omega H(w_1^*)}$. Then $\mathbb{C}$ sends $T_{w_c^*}$ to $\mathbb{A}$.
- **Phase2:** Consistent with Phase1.
- **Guess:** The $\mathbb{A}$ outputs the guessed answer $\tau^{'}$. If $\tau^{'} = \tau$, then $\mathbb{A}$ wins; otherwise $\mathbb{A}$ failed.

This is obtained by the above game. Probability that the attacker wins the keyword indistinguishability game is equal to the probability of the $\mathbb{C}$ breaking the *BDDH* problem. If T_w is computable, our scenario is in a real setting. If T_w is computationally unavailable, $\mathbb{A}$ cannot guess $\tau^{'}$. We prove the indistinguishability of the keyword and that $\mathbb{A}$ does not have any advantage to win the game.

□

Table 2 Functionality Comparison

Schemes	F1	F2	F3	F4	F5
[14]	Multiple	✓	✓	✓	×
[24]	Multiple	✓	✓	×	✓
[23]	Single	×	✓	×	×
Ours	Multiple	✓	✓	✓	✓

[1] F1:Authority; F2:Forward Security; F3:Backward Security; F4:Public Update; F5:Preventing Server and User Collusion.

5 Performance Analysis

We compare our schemes with respect to functionality, computational cost and storage size from the perspective of theoretical analysis and practical efficiency evaluation. Based on the consumption at different stages, we made a comparison of the performance of our scheme with the current schemes [14, 23, 24]. The notations in the chart are defined as follows: $|G|$ and $|G_T|$ represent respectively the lengths of the elements in $\mathcal{G}$ and G_T, and t_P, t_E, t_{E_T} are denoted by the bilinear pair operations, exponential operations in G and exponential operations in G_T. Define one data owner, multiple data users, and compare under single keyword encryption and search.

5.1 Functionality Analysis

Table 2 shows the functionality comparison of this paper's scheme with the frontier schemes [14, 23, 24]. As can be seen from the table, the scheme of this paper and scheme [14, 24] are implemented under multiple authorization authorities to prevent the single point of failure problem. Scheme [23] does not update the ciphertext stored at the server side, hence don't achieve forward security of the ciphertext. Schemes [14] and our scheme revoke the user's attributes using the KUNode algorithm, where the update key is publicly available. Scheme [24] and scheme [23] achieve backward security by updating the version key, where the version key needs to be delivered through a secure channel. Scheme [24] relies on smart contracts and consensus algorithms in the blockchain to achieve the verification of the user's identity, thus preventing the complicity between the server and the revoked user. The scheme in this paper employs public authentication to ensure the authenticity of the user's authorization without exposing any privacy.

5.2 Computational Analysis

Compared to related works [14, 23, 24], we analyze our computational overhead in terms of index generation, trapdoor generation, encryption, verification and user decryption phases. In the index generation step, these schemes have the similar computation cost in $2lt_E$, where l denotes the row in the access policy and t_E denotes the exponential operations in G. In the trapdoor genereation step, we have obvious advantage over [23] and [24] for without using bilinear paring. It seems that our scheme is more expensive in authentication, since it uses user attributes and implements user privilege verification in this step. We show the computation comparison in table 3.

Table 3 Computational Analysis.

Schemes	Trapdoor	Encrypt	Verify	Decrypt
[14]	$(5\left\|S\right\|+N)\left\|P\right\|t_E$	$(\left\|N_t\right\|\left\|b_\sigma\right\|+1)lt_E$	$-$	$t_{E_T}+2t_E$
[24]	$\left\|S\right\|t_E+t_P$	$5lt_E+t_{E_T}$	t_E	t_{E_T}
[23]	$(\left\|S\right\|+3)t_E+(\left\|S\right\|+2)t_{E_T}+2t_P$	$(n_{or}+1)t_P$	$(\left\|S\right\|+2)t_E+2t_P$	$-$
Ours	$(3\left\|S\right\|+N)\left\|P\right\|t_E+2t_E$	$(\left\|N_t\right\|\left\|b_\sigma\right\|+1)lt_E+(l+1)t_{E_T}$	$(\left\|S\right\|+1)t_P$	t_{E_T}

[1] Note. $|S|$: Size of the user attribute set S; N: Number of attribute authorities; l: row in the access policy; L: Complexity of access policies in decryption; $|P|$: Length of user binary tree path; $|N_t|$: Size of N_t in time binary tree; $|\,|b_\sigma|$: Length ofb_σ.

Table 4 Communication Analysis.

Schemes	F1	F2	F3	F4	F5
[14]	$4\left\|G\right\|$	$(d+3N)\left\|G\right\|+3N\left\|G_T\right\|$	$-$	$(2\left\|S\right\|\left\|P\right\|+2N)\left\|G\right\|+\left\|Z_q\right\|$	$((\left\|N_t\right\|d+4)l+2)\left\|G\right\|+(l+1)\left\|G_T\right\|$
[24]	$4\left\|G\right\|$	$3N\left\|G\right\|$	$2\left\|G\right\|$	$(\left\|S\right\|+2)\left\|G\right\|$	$3l\left\|G\right\|+\left\|G_T\right\|$
[23]	$(5+2N)\left\|G\right\|$	$-$	$\left\|G\right\|$	$(\left\|S\right\|+2)(\left\|G\right\|+\left\|G_T\right\|)$	$2l\left\|G_T\right\|+(n_{or}+1)\left\|Z_q\right\|$
Ours	$4\left\|G\right\|+2\left\|G_T\right\|$	$(d+N)\left\|G\right\|+N\left\|G_T\right\|$	$2\left\|G\right\|$	$(\left\|S\right\|\left\|P\right\|+2N)\left\|G\right\|$	$(\left\|N_t\right\|d+3)l\left\|G\right\|+(l+1)\left\|G_T\right\|$

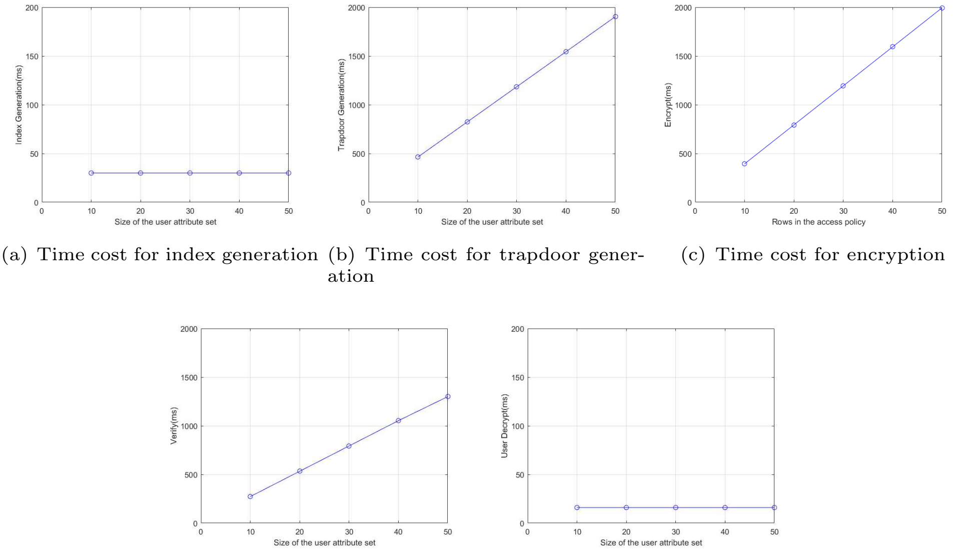
[2] Note. F1:Public Parameter Size; F2:Public Parameter Size; F3:Index Size; F4:Trapdoor Size; F5:CT Size.
[3] Note. n_{or}: Number of "OR" doors

5.3 Communication Analysis

As shown in the communication overhead in Table 4, we compare this paper's scheme with scheme [14, 23, 24] based on the size of the group elements. Because scheme [23] uses a single authorization authority, the communication cost of our scheme is larger than scheme [23, 24] and slightly smaller than scheme [14]. As shown in Table 4, the size of the keyword index of this paper's scheme is larger than that of scheme [23], because this paper implements a keyword index using the public parameter to update the keyword index. Comparing the communication cost of the trapdoor, our scheme is larger than scheme [23, 24] and slightly smaller than scheme [14]. This paper's scheme realizes the verification of user privileges without disclosing user privacy, and scheme [14] realizes the traceability of malicious users. After the function of preventing the server and malicious users from colluding, our scheme is not much different from scheme [14] in terms of the size of the ciphertext.

5.4 Experimental Analysis

In order to demonstrate the effectiveness of our scheme, we performed data evaluation with real data sets. We run our experiments using the JAVA language on a Lenovo AMD A8-6410 APU using AMD Radeon R5 Graphics 2.00GHz and 8GB RAM with Windows 10. The PBC-based cryptographic library was simulated and tested in terms



(a) Time cost for index generation (b) Time cost for trapdoor generation (c) Time cost for encryption

(d) Time cost for verify (e) Time cost for user decryption

Fig. 3 Actual performance.

of index generation time, trapdoor generation time, encrypt time, verify time, and user decrypt time. Furthermore, we set the depth of the time-binary tree to $d = 5$ for the real lab experiments, where $|b_\sigma| = 2$. In addition, to validate the performance of these solutions, we set the size of the set of employee attributes to $|S| \in [10, 50]$ and the number of attribute authorizations to $N = 8$. As shown in Figure 3(a), the actual computational cost in index generation remains constant. As shown in Figure 3(b), when the fixed path length is $|P| = 3$. The actual computational cost of generating trapdoors grows in a linear scale with the size of the set of employee attributes $|S|$. It is because they require additional computational effort to generate $|P|$ associated key elements for backward security. As shown in Figure 3(c), we set the behavior of the access policy $l \in [10, 50]$ to evaluate the impact of l on the computational cost of customer encryption and set the size of $|N_t| = 4$. When fixing the coefficient $|N_t|$, the actual computational cost of encryption increases linearly with l. The actual computational cost of encryption grows with l linearly. As shown in Figure 3(d), the actual computational cost of authentication increases linearly with the size of the employee attribute set $|S|$. As shown in Figure 3(e), the actual computational cost of employee decryption is almost constant.

6 Conclusion

We propose a secure and flexible authorization data sharing scheme for smart grid to solve the problem of attribute revocation where servers and malicious electricity company employees may collude to obtain customer data by using a means of publicly verifying the validity of electricity company employee trapdoors. Experimental simulations show that the scheme achieves secure, fine-grained attribute revocation and flexible ciphertext authorization search with less computation and communication overhead. The research in this paper aims to provide a new way of considering secure reversible prevention of server and revocation user collusion in smart grids. In the next step, we extend the scheme to a multi-keyword search, which improves the search precision.

Declarations

- Funding. This study was funded by ANatural Science Foundation of Sichuan, China (2023NSFSC1400), Chengdu Science and Technology Program (2021-YF08-00151-GX).

- Conflict of interest/Competing interests. We confirm that there are no conflicts of interest associated with thsubmission of this manuscript.
- Ethics approval. All authors read and approved the final manuscript.
- Consent to participate. All authors consent to participate.
- Consent for publication. All authors consent for publication.
- Availability of data and materials. We make sure that all data and materials support their published claims and comply with field standards.
- Code availability. We make sure that code supports their published claims and comply with field standards.
- Authors' contributions. All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Yawen Feng and Shengke Zeng. The first draft of the manuscript was written by Yawen Feng and all authors commented on previous versions of the manuscript.

References

[1] Kirmani, S., Mazid, A., Khan, I.A., Abid, M.: A survey on iot-enabled smart grids: Technologies, architectures, applications, and challenges. Sustainability **15**(1) (2023)

[2] Jafari, M., Kavousi-Fard, A., Chen, T., Karimi, M.: A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future. IEEE Access **11**, 17471–17484 (2023) https://doi.org/10.1109/ACCESS.2023.3241588

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

[4] Zhao, M., Ding, Y., Tang, S., Liang, H., Wang, H.: A blockchain-based framework for privacy-preserving and verifiable billing in smart grid. Peer-to-Peer Networking and Applications **16**(1), 142–155 (2023)

[5] Mehta, P.J., Parne, B.L., Patel, S.J.: Se-lakaf: Security enhanced lightweight authentication and key agreement framework for smart grid network. Peer-to-Peer Networking and Applications **16**(3), 1513–1535 (2023)

[6] Cao, Y., Li, S., Lv, C., Wang, D., Sun, H., Jiang, J., Meng, F., Xu, L., Cheng, X.: Towards cyber security for low-carbon transportation: Overview, challenges and future directions. Renewable and Sustainable Energy Reviews **183**, 113401 (2023)

[7] Egide, N., Li, F.: Hap-sg: Heterogeneous authentication protocol for smart grid. Peer-to-Peer Networking and Applications, 1–15 (2023)

[8] Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy. SP 2000, pp. 44–55 (2000). https://doi.org/10.1109/SECPRI.2000.848445

[9] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Advances in Cryptology - EUROCRYPT 2004, pp. 506–522 (2004)

[10] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: International Conference on the Theory and Application of Cryptographic Techniques (2005). https://api.semanticscholar.org/CorpusID:10137076

[11] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. IACR Cryptol. ePrint Arch. **2006**, 309 (2006)

[12] Eltayieb, N., Elhabob, R., Hassan, A., Li, F.: An efficient attribute-based online/offline searchable encryption and its application in cloud-based reliable smart grid. Journal of Systems Architecture **98**, 165–172 (2019) https://doi.org/10.1016/j.sysarc.2019.07.005

[13] Zhang, X., Mu, D., Zhao, J.: Attribute-based keyword search encryption for power data protection. High-Confidence Computing, 100115 (2023)

[14] Zhang, J., Ma, J., Yang, Y., Liu, X., Xiong, N.N.: Revocable and privacy-preserving decentralized data sharing framework for fog-assisted internet of things. IEEE Internet of Things Journal **9**(13), 10446–10463 (2022)

[15] Ma, K., Song, G., Zhou, Y., Xu, R., Yang, B.: An efficient identity authentication protocol with revocation, tracking and fine-grained access control for electronic medical system. Computer Standards Interfaces **88**, 103784 (2023)

[16] Li, J., Zhang, T.: Power data attribution revocation searchable encrypted cloud storage. In: 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), pp. 579–582 (2023). IEEE

[17] Yang, Y., Deng, R.H., Guo, W., Cheng, H., Luo, X., Zheng, X., Rong, C.: Dual traceable distributed attribute-based searchable encryption and ownership transfer. IEEE Transactions on Cloud Computing **11**(1), 247–262 (2023)

[18] Ge, C., Susilo, W., Liu, Z., Baek, J., Luo, X., Fang, L.: Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds. IEEE Transactions on Dependable and Secure Computing, 1–12 (2023) https://doi.org/10.1109/TDSC.2023.3265979

[19] Wang, M., Miao, Y., Guo, Y., Huang, H., Wang, C., Jia, X.: Aesm2 attribute-based encrypted search for multi-owner and multi-user distributed systems. IEEE Transactions on Parallel and Distributed Systems **34**(1), 92–107 (2023) https://doi.org/10.1109/TPDS.2022.3216320

[20] Niu, S., Hu, Y., Zhou, S., Shao, H., Wang, C.: Attribute-based searchable encryption in edge computing for lightweight devices. IEEE Systems Journal **17**(3), 3503–3514 (2023) https://doi.org/10.1109/JSYST.2023.3283389

[21] Abbou, R.B., Mrabti, F., Ghoubach, I.E.: Efficient and secure data sharing with outsourced decryption and efficient revocation for cloud storage systems. International Journal of Security and Networks **14**(3), 133 (2019)

[22] Ghopur, D., Ma, J., Ma, X., Hao, J., Jiang, T., Wang, X.: Puncturable key-policy attribute-based encryption scheme for efficient user revocation. IEEE Transactions on Services Computing, 1–12 (2023) https://doi.org/10.1109/TSC.2023.3303368

[23] Sultan, N.H., Kaaniche, N., Laurent, M., Barbhuiya, F.A.: Authorized keyword search over outsourced encrypted data in cloud environment. IEEE Transactions on Cloud Computing **10**(1), 216–233 (2022)

[24] Yu, J., Liu, S., Xu, M., Guo, H., Zhong, F., Cheng, W.: An efficient revocable and searchable ma-abe scheme with blockchain assistance for c-iot. IEEE Internet of Things Journal **10**(3), 2754–2766 (2023) https://doi.org/10.1109/JIOT.2022.3213829

[25] Edemacu, K., Jang, B., Kim, J.W.: Cescr: Cp-abe for efficient and secure sharing of data in collaborative ehealth with revocation and no dummy attribute. PloS one **16**(5), 0250992 (2021)

[26] Austin, A.J.: Sharing phr data in cloud using sigmoid key and median support signature-based cryptosystem. Wireless Personal Communications **124**(4), 3549–3565 (2022)

[27] Wei, J., Liu, W., Hu, X.: Secure and efficient attribute-based access control for multiauthority cloud storage. IEEE Systems Journal, 1–12 (2016)

[28] Liu, X., Lu, T., He, X., Yang, X., Niu, S.: Verifiable attribute-based keyword search over encrypted cloud data supporting data deduplication. IEEE Access **8**, 52062–52074 (2020) https://doi.org/10.1109/ACCESS.2020.2980627