High-Confidence Computing 3 (2024) 100115

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Research

Attribute-based keyword search encryption for power data protection

Xun Zhang [a,b,*], Dejun Mu [a,b], Jinxiong Zhao [a,b]

[a] School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China
[b] Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China

article info

Article history:
Received 14 November 2023
Revised 2 January 2024
Accepted 5 February 2024

Keywords:
Attribute-based encryption
Searchable encryption
Keyword search
Power grid data

abstract

To protect the privacy of power data, we usually encrypt data before outsourcing it to the cloud servers. However, it is challenging to search over the encrypted data. In addition, we need to ensure that only authorized users can retrieve the power data. The attribute-based searchable encryption is an advanced technology to solve these problems. However, many existing schemes do not support large universe, expressive access policies, and hidden access policies. In this paper, we propose an attribute-based keyword search encryption scheme for power data protection. Firstly, our proposed scheme can support encrypted data retrieval and achieve fine-grained access control. Only authorized users whose attributes satisfy the access policies can search and decrypt the encrypted data. Secondly, to satisfy the requirement in the power grid environment, the proposed scheme can support large attribute universe and hidden access policies. The access policy in this scheme does not leak private information about users. Thirdly, the security analysis and performance analysis indicate that our scheme is efficient and practical. Furthermore, the comparisons with other schemes demonstrate the advantages of our proposed scheme.

© 2024 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

With the development of information technology, the power grid system has been collecting a large amount of power data constantly [1]. These data involve a lot of user privacy and have an important value in data analysis [2]. Therefore, how to store and process these sensitive data in the power grid system has become a very important issue. In recent years, due to the strong computing and storage capabilities of cloud computing [3,4], people are willing to store data on cloud servers rather than local servers, which can reduce the burden of managing large amounts of data and improve the efficiency of the data processing. However, because cloud service providers are not fully trusted, there are some threats in the cloud environment, such as data leakage, wrong access decisions, external malicious attacks, etc. It means that outsourcing data directly to cloud servers is not a good idea, which may lead to privacy disclosure problems. Especially for power data, once the sensitive data is disclosed, it will hurt people's interests and even national security. Therefore, to protect the privacy of power data, we usually encrypt data before outsourcing it to cloud servers. Although using encryption technology protects data privacy, it brings a new problem: how to search over the encrypted data.

To address the above issues, searchable encryption (SE) [5] has been proposed. Using searchable encryption can achieve the retrieval operation over ciphertext without decryption. In the searchable encryption scheme, data owners outsource encrypted data to cloud servers firstly. Data users can achieve keyword search over encrypted data by using the search trapdoor that can be generated by data users' private keys and search keywords. Note that the search trapdoor does not leak any useful information about the search keyword. Then, the cloud servers take the trapdoor as the input, execute the retrieval algorithm, and send the ciphertext associated with keywords to data users. Finally, data users execute the decryption algorithm to decrypt the ciphertext.

In addition, fine-grained access control is also required for the power grid system. There are great varieties of users in the power grid system, and data owners usually want their power data to be retrieved by other authorized users. However, in the traditional searchable encryption scheme, data owners cannot define access policies by themselves, which leads to data owners cannot specify which users can retrieve their outsourced data. Fortunately, attribute-based encryption (ABE) [6] can address this problem. ABE is considered to be an important encryption technology that can achieve fine-grained access control. Generally, scholars classify ABE into two different types [7], which are key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). In the KP-ABE scheme, secret keys are associated with access policies, while ciphertext are

* Corresponding author.
E-mail address: zhangxunsf@mail.nwpu.edu.cn (X. Zhang).

https://doi.org/10.1016/j.hcc.2024.100115
2667-2952/© 2024 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

associated with attributes. On the contrary, in the CP-ABE scheme, ciphertext are associated with access policies and secret keys are associated with attributes. Therefore, in the CP-ABE scheme, data owners can define their own access policies and encrypt data according to the policies, and only users whose attributes satisfy the access policies can decrypt the data. This is the reason that CP-ABE is more suitable for access control than KP-ABE.

By combining SE and ABE technologies, scholars proposed the concept of attribute-based searchable encryption (ABSE) [8]. We can achieve both encrypted data retrieval and fine-grained access control by using ABSE. However, most of the existing schemes do not support large universe requirement and expressive access policies. And these two characteristics are necessary for the power grid system. In addition, because the access policy will be stored on the cloud servers, how to prevent privacy leakage from the access policy is an important problem.

In this paper, we propose an attribute-based keyword search encryption scheme for power data protection. Our proposed scheme has some important strengths, which make the scheme is suitable for application in the power grid system. Our scheme can support encrypted data retrieval and achieve fine-grained access control. The power data is encrypted and outsourced to cloud servers. And the encrypted data can be searched by using trapdoor which does not reveal any useful information about the search keyword. Data users can search and access those power data based on their attributes. In addition, in our scheme, we use CP-ABE to achieve access control. So, data users can generate secret keys based on their attributes. It makes our scheme more reasonable for application. Furthermore, our scheme can support large universe, expressive access policies, and hidden access policies. In our scheme, the size of PK is constant rather than increasing linearly with attributes, which make our scheme is more suitable for large universe system. In addition, our scheme is based on LSSS structure that can support expressive access policies. In our scheme, each attribute has two parts: attribute name and attribute value. The attribute name is stored on the cloud servers publicly, and the attribute value is hidden. Therefore, our scheme can hidden access policies partially. Our main contributions are as follows:

Firstly, we propose an attribute-based keyword search encryption scheme for power data protection. This scheme can achieve both encrypted data retrieval and fine-grained access control. In this scheme, only users whose attributes satisfy the access policies can search the encrypted data.

Secondly, to satisfy the requirement for searchable encryption in the power grid system, we improve Ge et al.'s scheme [9] and make the scheme support large universe requirement and expressive access policies. Furthermore, to protect users' privacy, we hide the access policy partially.

Thirdly, we further analyze the security and evaluate performance of the proposed scheme. In addition, comparisons of other related schemes demonstrate the advantages of our proposed scheme.

The remainder of this paper is organized as follows. We describe the related work in Section 2. And we introduce some preliminary knowledge in Section 3. In Section 4, we define the system model and the procedure of this system. And then, we describe the detailed construction of the proposed scheme in Section 5. Section 6 provides the security analysis, performance analysis and comparisons with other schemes. At last, we conclude this paper in Section 7.

2. Related work

2.1. Attribute-based encryption schemes

Sahai et al. [6] introduced the concept of ABE in 2005 to support the error-tolerance property of the identity-based encryption. This scheme views an identity as a set of descriptive attributes. In 2007, Bethrncourt et al. [10] proposed the CP-ABE scheme firstly. In this scheme, the user's private key is associated with a set of attributes, and the ciphertext of the data is associated with access policy. Only users whose attributes satisfy the access policy can decrypt the data. Waters et al. [11] proposed a CP-ABE scheme based on the linear secret sharing scheme (LSSS). Compared with the schemes using other access structures, the LSSS structure in this scheme can support expressive access policies. Notably, using some techniques [12,13] can convert any access tree structure into an LSSS representation. Since then, many attribute-based encryption schemes using LSSS structure have been proposed. Lai et al. [14] proposed expressive CP-ABE with partially hidden access structures. In this scheme, each attribute is defined in two parts: attribute name and attribute value. The attribute name is public and the attribute value is hidden. In addition, there are some schemes [15,16] also achieve partially hidden access polices, but those schemes are based on the "AND gate" access structure, which limits the expressivity of the access policy. Zhang et al. [17] improved scheme [14] into large universe, and improved the efficiency of the decryption test part, making it more suitable for the application environment of intelligent medical treatment. Yang et al. [18] proposed a fine-gained big data access control scheme with fully hidden access policy. And in scheme [18], the authors designed an attribute bloom filter to evaluate and locate whether an attribute is in the access policy.

2.2. Searchable encryption schemes

Song et al. [5] first introduced searchable encryption. Song et al.'s scheme is based on sequential scanning which refers to a search over the entire encrypted document. And this search method makes the scheme inefficient. Boneh et al. [19] proposed the notion of public key encryption with keyword search (PEKS). Then, how to improve efficiency, security, and query expressiveness are discussed extensively in many schemes. Park et al. [20] proposed the conjunctive-based searchable encryption that can support conjunctive keyword search. But this scheme has low search efficiency because of pairing operations. Li et al. [21] proposed a fuzzy keyword search scheme that suffers from high computation and storage costs. In recent years, due to the requirement of access control, some attribute-based searchable encryption schemes have been proposed. Li et al. [22] proposed two attribute-based keyword search and data access control schemes that use CP-ABE to control the access policy and KP-ABE to control the search policy. Li et al. [23] proposed a keyword-searchable attribute-based encryption scheme based on lattice cryptography. Lattice cryptography is an advanced technology that can protect the proposed scheme against quantum attacks. However, the authors use KP-ABE in the proposed system, which will lead to extra communication costs. Compared to KP-ABE, CP-ABE is more suitable for access control. Xu et al. [24] also proposed a searchable encryption scheme based on lattice. The authors use blockchain technology to ensure the traceability of keyword retrieval process and maintain the credibility of search results. Ge et al. [9] proposed a ciphertext-policy attribute-based mechanism with keyword search. The proposed solution utilizes the scheme [11] as the basic component, and can support attribute-based keyword search and keyword update. However, in this scheme, the public key size grows linearly with the number of attributes. This property is not suitable for search encryption systems that have a large scale of the attribute universe. Furthermore, this scheme does not support access policies hidden. Miao et al. [25] proposed a privacy-preserving CP-ABKS system with hidden access policy. However, similar to scheme [9], this scheme also does not support large universe.

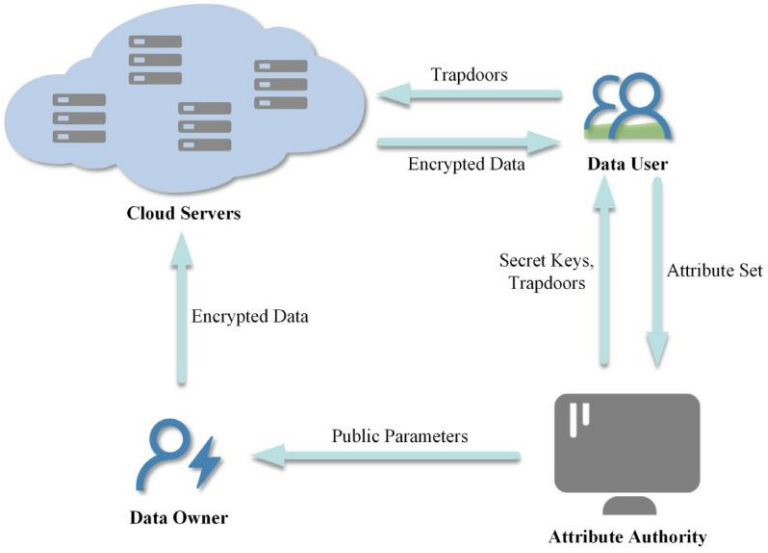

Fig. 1. System model.

3. Preliminaries

3.1. Bilinear pairing

Let G and G_T are cyclic groups of order p, a bilinear map is a map $e : G \times G \rightarrow G_T$ with the following properties:

(1) Bilinear: $\forall u, v \in G, a, b \in Z_p, e(u^a, v^b) = e(u, v)^{ab}$.
(2) Non-degenerate: $\exists g \in G$ such that $e(g, g) \neq 1$.
(3) Computable: e can be computed.

3.2. Linear secret sharing schemes

Our proposed scheme will employ LSSS [26]. A secret sharing scheme Π over a set of parties P is called linear over Z_p if:

(1) The shares for each party form a vector over Z_p.
(2) There exists a matrix $\mathbf{A}$ with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, the i row of $\mathbf{A}$ is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \ldots, \ell\}$ to P). When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in Z_p$ is the secret to be shared, and $r_2, \ldots, r_n \in Z_p$ are randomly chosen, then $\mathbf{A}v$ is the vector of ℓ shares of the secret s according to Π. The share $(\mathbf{A}v)_i$ belongs to party $\rho(i)$.

It is shown in [26] that every linear secret-sharing scheme according to the above definition also enjoys the linear reconstruction property, defined as follows: Suppose that Π is an LSSS for access structure A. Let $S \in A$ be an authorized set, and let $I \subset \{1, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. There exist constants $\{\omega_i \in Z_p\}_{i \in I}$ such that if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, these constants ω_i can be found in time polynomial in the size of the share-generating matrix $\mathbf{A}$. For any unauthorized set, no such constants exist.

3.3. Attribute and access policy

In the CP-ABE scheme, ciphertext are associated with access policies and secret keys are associated with attributes. The user can decrypt the data only if user's attribute set satisfies the access policies. Similar to Lai et al.'s scheme [14], each attribute in this paper includes two parts: attribute name and attribute value. The attribute name is public and the attribute value is hidden. We assume that there are n categories of attribute names and every user has n attribute values with each attribute value belonging to a different attribute name. Furthermore, let i denote the attribute name of the ith category attribute. And each attribute value can be described as a binary string $\{0, 1\}^*$.

We express the access policy by $A = (\mathbf{A}, \rho, T)$, where $\mathbf{A}$ is $\ell \times n$ share-generating matrix, ρ is a map from each row of $\mathbf{A}$ to an attribute name (i.e., ρ is a function from $\{1, \ldots, \ell\}$ to $\{1, \ldots, n\}$), T can be parsed as $(t_{\rho(1)}, \ldots, t_{\rho(\ell)})$ and $t_{\rho(i)}$ is the value of attribute $\rho(i)$ specified by the access formula.

4. System overview

4.1. System model

In this section, we first present the model of an attribute-based keyword search encryption scheme for power data protection as shown in Fig. 1. There are four entities in our proposed scheme, namely data owner, data user, cloud server, and attribute authority. The entities are defined as follows.

Data Owner: Data owners refer to the users who collect power data in the power grid system. And data owners are responsible for outsourcing their private data into the cloud servers. They can encrypt their data with access policies and keywords. Only users whose attributes satisfy the access policies can search and decrypt the encrypted data.

Data User: Data users are responsible for searching the data from cloud servers. With the help of the attribute authority, data users can generate secret keys by using their own attributes. And then, data users can use the secret keys to generate search trapdoors to perform keyword search over encrypted data. In addition, data users can decrypt the ciphertext by using their secret keys.

Cloud Server: Cloud servers are responsible for storing encrypted data and executing search operations on behalf of data users. Data users only need to send the search trapdoors to cloud servers without considering the details of the search process. And cloud servers cannot know any useful information on the search keyword.

Attribute Authority: The attribute authority is responsible for managing attributes in the system. In addition, the attribute

authority can not only generate secret keys by using the data user's attributes but also generate search trapdoors by using secret keys and keywords. Note that the attribute authority is completely trusted in the proposed system.

4.2. The procedure of system

The proposed system has the following four phases:

System Initialization: In this phase, the attribute authority initializes attribute space and assigns attributes to system users. Then, the attribute authority runs the Setup algorithm to generate the system public parameters and master secret key. The public parameters will be sent to data owners for data encryption and the master secret key will be kept in attribute authority for secret key generation.

Ciphertext Upload: Data owners can encrypt their power data with access policies and keywords. The access policies are defined by data owners themselves. This means that data owners can determine which users have the right to search and access the encrypted data. After encrypting data, data owners upload the ciphertext of power data and access policies to the cloud servers. Note that data owners only upload the attribute names in the access policies to the cloud servers. The attribute name can help users to evaluate whether their attributes are in the access policy and do not leak information about users.

Ciphertext Retrieval: In this phase, data users send their attribute set and the search keywords to the attribute authority. The attribute authority generates the secret keys based on the users' attributes first and generates the search trapdoors based on secret keys and the search keywords. The trapdoors will be returned to data users for ciphertext search query. After data users send the trapdoors to cloud servers, the Test algorithm will be run. This algorithm will compare whether the keywords in the trapdoors and the keywords in the ciphertext are the same. Then, the cloud servers return the search results to data users.

Data Decryption: In the final phase, data users can obtain the secret keys from the attribute authority and the ciphertext from cloud servers. By running the Decrypt algorithm, data users can decrypt the ciphertext and get the power data that they want to retrieve.

5. Construction of the proposed scheme

The attribute-based keyword search encryption scheme consists of six algorithms: Setup, KeyGen, Encrypt, TrapdoorGen, Test, and Decrypt.

Setup(λ): This algorithm takes a security parameter λ as input. And let G and G_T be cyclic groups of prime order p, and $e: G \times G \to G_T$ be a bilinear map. Then it chooses a generator $g \in G$, $\alpha, a, b, c \in Z_p^*$, and $f, g^{\sim} \in G$, and computes $f_1 = g^c$, $f_2 = g^b$. It also generates four hash functions: $H_1: \{0,1\}^* \to G$, $H_2: G_T \to \{0,1\}^*$, $H_3: \{0,1\}^* \to Z_p^*$, $H_4: \{0,1\}^* \times G_T \to Z_p^*$. Next, the public parameters are published as $PK = \{e(g,g)^{\alpha}, g^a, g^{\sim}, f, f_1, f_2, H_1, H_2, H_3, H_4\}$, and the master secret key is $MSK = \{g^{\alpha}, a, b\}$.

KeyGen(MSK, S): This algorithm takes the master secret key MSK and the data user's attribute set $S = \{att_1, \ldots, att_n\}$ as inputs. Note that att_i is the value of attribute i and each attribute value in the set S can be described as a binary string $\{0,1\}^*$. Then, it chooses $t, r \in Z_p^*$, and computes the secret key sk_S:

$$K = g^{\alpha} f^t, L = g^t, V = g^{(ac-r)/b}, Y = g^r, Z = g^{\sim r}, \\ \forall i \in [1,n], \{K_i = H_1(att_i)^t, Y_i = H_1(att_i)^r\}. \tag{1}$$

Encrypt($m, KW, (M, \rho, T)$): This algorithm takes as inputs data m, keyword KW, and the data owner's access policy (M, ρ, T), It chooses $R \in G_T$ randomly, and computes $s = H_4(m, R)$. Then, this algorithm constructs two random vectors $\vec{v} = (s, v_2, \ldots, v_n)$ and $\vec{v'} = (s_2, v'_2, \ldots, v'_n)$, where $s_2, v_2, v'_2, \ldots, v_n, v'_n$ are choosed randomly from Z_p^*. For $i \in [1, \ell]$, it calculates $\lambda_i = \vec{v} \cdot M_i$ and $\lambda'_i = \vec{v'} \cdot M_i$, where M_i is the vector related to the ith row of M.

Then, it chooses $s_1 \in Z_p^*$ randomly and computes:

$$C_0 = m \oplus H_2(R), C = R \cdot e(g,g)^{\alpha s}, C' = g^s, \\ \forall i \in [1,\ell], C_i = f^{\lambda_i} H_1(att_{\rho(i)})^{-s}, \tag{2}$$

$$W_0 = g^{a(s_1+s_2)} f_2^{s_1 H_3(KW)}, W_1 = f^{s_1}, W_2 = f^{s_2}, D = g^{s_2}, \\ \forall i \in [1,\ell], E_i = g^{\sim \lambda'_i} H_1(att_{\rho(i)})^{-s_2}, \tag{3}$$

$$E = H_1(C_0, C, C', D, \{C_i, E_i\}_{i\in[1,\ell]}, W_0, W_1, W_2)^s. \tag{4}$$

And output the ciphertext:

$$CT = (C_0, C, C', D, \{C_i, E_i\}_{i\in[1,\ell]}, W_0, W_1, W_2, E). \tag{5}$$

TrapdoorGen(sk_S, KW'): This algorithm takes as inputs secret key sk_S and the search keyword KW'. It chooses $\gamma \in Z_p^*$ and computes: $\tau_1 = (g^a f_2^{H_3(KW')})^{\gamma}$, $\tau_2 = f^{\gamma}$, $\tau_3 = V^{\gamma}$, $Y' = Y^{\gamma}$, $Z' = Z^{\gamma}$. And, for $x \in S$, it calculates $Y'_x = Y^{\gamma}_x$.

And output the trapdoor $\tau = (\tau_1, \tau_2, \tau_3, Y', Z', \{Y'_x\}_{\forall x \in S})$.

Test(CT, τ): This algorithm takes as inputs the ciphertext CT and the trapdoor τ. Suppose that user's attribute set S satisfies the access policy and let $I \subseteq \{1, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. There exist constants $\{\omega'_i \in Z_p^*\}_{i\in I}$ and $\sum_{i\in I} \omega'_i \lambda'_i = s_2$. It computes:

$$F = e(Y'Z', D) / \left(\prod_{i\in I} (e(Y', E_i) \cdot e(D, Y'_{\rho(i)}))^{\omega'_i}\right). \tag{6}$$

Then, if $KW = KW'$, the equation $e(W_1, \tau_1)e(W_2, \tau_3)F = e(W_0, \tau_2)$ hold, it returns 1. Otherwise returns 0.

Decrypt(sk_S, CT): This algorithm takes as input a ciphertext CT and a secret key sk_S. If the data user's attributes satisfy the data owner's access policy, it can find constants $\{\omega_i \in Z_p^*\}_{i\in I}$ and $\sum_{i\in I} \omega_i \lambda_i = s$, where $I = \{i : \rho(i) \in S\}$. It computes:

$$\prod_{i\in I} \frac{e(K, C')}{e(C_i, L)^{\omega_i} \cdot e(C', K_{\rho(i)})^{\omega_i}} = e(g,g)^{\alpha s}. \tag{7}$$

Then, it computes $R = C / e(g,g)^{\alpha s}$, $m = C_0 \oplus H_2(R)$ and $s = H_4(m, R)$, and checks $C' = g^s$, $E = H_1(C_0, C, C', D, \{C_i, E_i\}_{i\in[1,\ell]}, W_0, W_1, W_2)^s$.

If these equations hold, it recovers the data as m. Otherwise, it outputs $\perp$ to denote that the decryption fails.

The correctness of the proposed scheme is verified as follows.

Firstly, we verify the correctness of the Test algorithm. In the Test algorithm, we perform Eq. (8).

$$\begin{aligned} F &= e(Y'Z', D) / \left(\prod_{i\in I} (e(Y', E_i) \cdot e(D, Y'_{\rho(i)}))^{\omega'_i}\right) \\ &= \frac{e(g^{r\gamma} \cdot g^{\sim r\gamma}, g^{s_2})}{\prod_{i\in I} (e(g^{r\gamma}, g^{\sim \lambda'_i} H_1(att_{\rho(i)})^{-s_2}) \cdot e(g^{s_2}, H_1(att_{\rho(i)})^{r\gamma}))^{\omega'_i}} \\ &= \frac{e(g^{r\gamma}, g^{s_2}) e(g^{\sim r\gamma}, g^{s_2})}{\prod_{i\in I} (e(g, H_1(att_{\rho(i)}))^{-s_2 r\gamma} \cdot e(g, H_1(att_{\rho(i)}))^{s_2 r\gamma} \cdot e(g^{r\gamma}, g^{\sim \lambda'_i}))^{\omega'_i}} \\ &= \frac{e(g,g)^{s_2 r\gamma} e(g, g^{\sim})^{s_2 r\gamma}}{e(g, g^{\sim})^{r\gamma \sum_{i\in I} \lambda'_i \cdot \omega'_i}} \\ &= \frac{e(g,g)^{s_2 r\gamma} e(g, g^{\sim})^{s_2 r\gamma}}{e(g, g^{\sim})^{r\gamma s_2}} \\ &= e(g,g)^{s_2 r\gamma}. \end{aligned} \tag{8}$$

Then, we assume $KW = KW'$, and the correctness of the Test algorithm can be verified by performing Eq. (9).

$$\begin{aligned}
&e(W_1, \tau_1)e(W_2, \tau_3)F \\
&= e(f_1^{s_1}, (g^a f_2^{H_3(KW')})^{\gamma}) \cdot e(f_2^{s_2}, (g^{(ac-r)/b})^{\gamma}) \cdot e(g,g)^{s_2 r\gamma} \\
&= e(f_1, g)^{s_1 a\gamma} \cdot e(f_1, f_2)^{s_1 \gamma H_3(KW')} \cdot e(g, f_1)^{s_2 a\gamma} \cdot e(g,g)^{-s_2 r\gamma} \\
&\quad \cdot e(g,g)^{s_2 r\gamma} \\
&= e(g, f_1)^{a\gamma(s_1+s_2)} \cdot e(f_1, f_2)^{s_1 \gamma H_3(KW')} \\
&= e(g^{a(s_1+s_2)} f_2^{s_1 H_3(KW')}, f_1^{\gamma}) \\
&= e(W_0, \tau_2).
\end{aligned} \tag{9}$$

Secondly, we verify the correctness of the Decrypt algorithm.

We assume the data user's attributes satisfy the data owner's access policy, and perform Eq. (10).

$$\begin{aligned}
&\frac{\prod_{i \in I} e(K, C')}{\prod_{i \in I} e(C_i, L)^{\omega_i} \cdot e(C', \prod_{i \in I} K_{\rho(i)}^{\omega_i})} \\
&= \frac{\prod_{i \in I} e(g^a f^t, g^s)^{\omega_i}}{\prod_{i \in I} e(f^{\lambda_i} H_1(att_{\rho(i)})^{-s}, g^t)^{\omega_i} \cdot e(g^s, \prod_{i \in I} H_1(att_{\rho(i)})^{\omega_i})} \\
&= \frac{e(g,g)^{as} \cdot e(f,g)^{ts}}{e(f,g)^{t \sum_{i \in I} \lambda_i \omega_i} \cdot \prod_{i \in I} e(H_1(att_{\rho(i)}), g)^{-st\omega_i} \cdot \prod_{i \in I} e(g, H_1(att_{\rho(i)}))^{st\omega_i}} \\
&= \frac{e(g,g)^{as} \cdot e(f,g)^{ts}}{e(f,g)^{ts}} \\
&= e(g,g)^{as}.
\end{aligned} \tag{10}$$

And, the data m can be decrypt by computing $R = C / e(g,g)^{as}$, $m = C_0 \oplus H_2(R)$. So, the correctness of the Decrypt algorithm can be verified.

6. Analysis of our scheme

6.1. Security analysis

The proposed scheme satisfies the security properties described as follows.

6.1.1. Secure retrieval over encrypt data

In our scheme, we encrypt power data before outsourcing it to the cloud servers. Because the data is encrypted on the cloud servers, malicious users cannot get useful information from the ciphertext if they do not have a correct secret key. In addition, outsourcing encrypted data to the cloud servers can reduce the burden of managing large amounts of power data. The power data is encrypted and stored on cloud servers, which protects the privacy of the power data. Furthermore, our proposed scheme over power grid data can achieve encrypted data retrieval by using the search trapdoor. And the trapdoor does not reveal any useful information about the search keyword. Therefore, our scheme can achieve secure storage and retrieval of the power data.

6.1.2. Fine-grained access control

In the proposed scheme, we use CP-ABE to achieve fine-grained access control. Data owners can encrypt data under their defined access policies. And data users can search and access those power data based on their attributes. Meanwhile, unauthorized users whose attributes do not satisfy access policies cannot search and decrypt the ciphertext successfully.

6.1.3. Hidden access policy

To decrypt the ciphertext, data users need to know which attributes are involved in the access policy. Therefore, the access policy must also be stored together on the cloud servers with the ciphertext. It leads to that malicious users can obtain some privacy information from the access policy on the cloud servers. To address this issue, in our scheme, each attribute has two parts: attribute name and attribute value. The attribute name is stored on the cloud servers publicly, and the attribute value is hidden. Hence, data users can know which attributes are involved in the access policy by the attribute name, and meanwhile, the attribute name does not leak private information about users.

6.1.4. Provable security

We can prove the security of our proposed scheme against chosen ciphertext attack (CCA) and chosen keyword attack (CKA) in the random oracle model. In [9], Ge et al. proved their scheme's security. Because our scheme is improved based on the scheme of Ge et al. more information on security proof can be found in [9]. And we do not discuss the details of security proof in this paper.

6.2. Performance analysis

In this section, we implement our proposed scheme on a laptop and evaluate the performance of the scheme. Our experiments are implemented on a 64-bit Windows 10 system with 2.90 GHz Intel(R) Core(TM) i5-6267U CPU and 8.00G RAM. And our experiments are based on Java language and the Java Pairing Based Cryptography Library (JPBC) of version 2.0.0 [27]. We use the type A parings, which are constructed on curve $y^2 = x^3 + x$. For each result, the experiment is repeated 20 times, and we use the average values to evaluate the performance. In Fig. 2, we show the implementation time of algorithms, including trapdoor generation algorithm, test algorithm, encryption algorithm, and decryption algorithm. The x-axis is the number of the data user's attributes and the y-axis is the implementation time of the algorithms. The results show that the implementation time increases with the growth of the number of attributes. The reason is that the consumed time of algorithms is increasingly affected by the growth of the number of attributes. And as illustrated, the encryption algorithm consumes more time than other algorithms. And with the growth of the number of attributes, the implementation time of trapdoor generation algorithm is gradually less than the other three algorithms. To further evaluate the performance of our proposed scheme, we compare the key generation time of our proposed scheme with some related schemes [25,28] in Fig. 3. We can observe that our scheme is more efficient than other schemes in key generation. Obviously, by implementing our proposed scheme, the results indicate that our scheme is efficient.

6.3. Comparisons with other schemes

In this section, we compare our scheme with other related schemes. We first compare the public parameters PK of our proposed scheme with Ge et al.'s scheme [9]. In this scheme, the Setup algorithm outputs $PK = \{e(g,g)^a, g^a, \tilde{g^{}}, f, f_1, f_2, Q, H_1, H_2, H_3, H_4, h_1, \ldots, h_{|U|}, SY\}$. For each i from attribute space U, the Setup algorithm chooses $h_i \in G$. It makes the public parameters size grows linearly with the number of attributes. And in this scheme, not all h_i will be used for encryption, only those h_i which are related to attributes in the access policies will be used. Therefore, to address this problem, we remove all h_i in the public parameters, which makes the size of PK constant rather than increasing linearly with attributes. The comparison of the size of PK is shown in Fig. 4. In addition, we compare the implementation time of Setup algorithm in Fig. 5. In Ge et al.'s scheme [9], the time of Setup algorithm increases with the growth of the number of attributes, because attributes affect the size of the public key. On the contrary, in our proposed scheme, the time

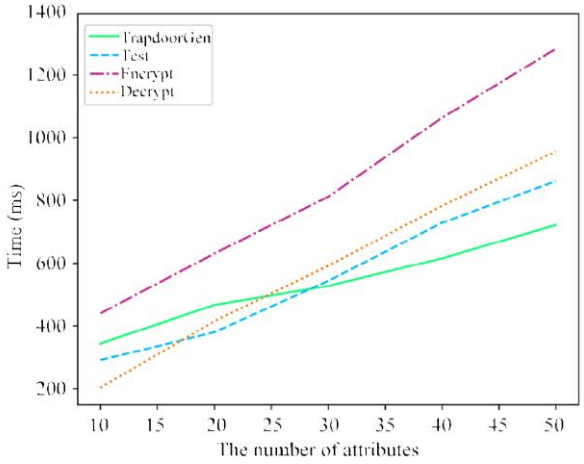

Fig. 2. The implementation time of algorithms.

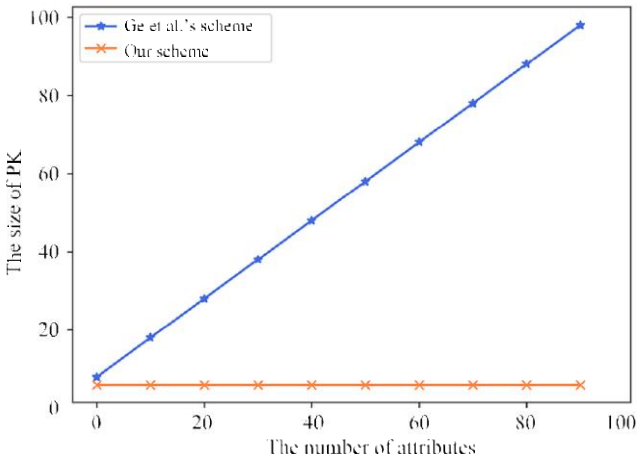

Fig. 4. The size of the public parameters.

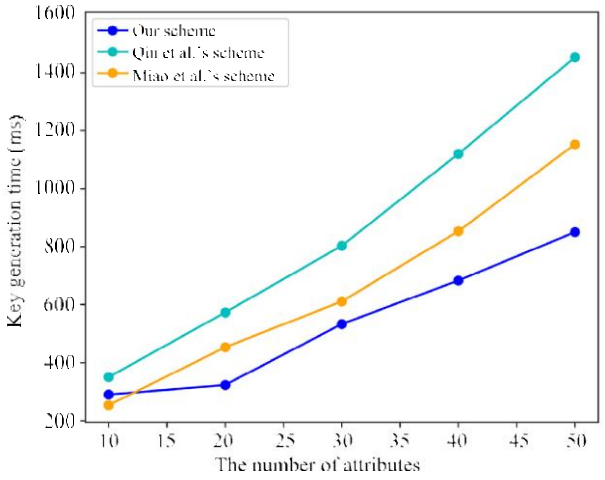

Fig. 3. The comparison of key generation time.

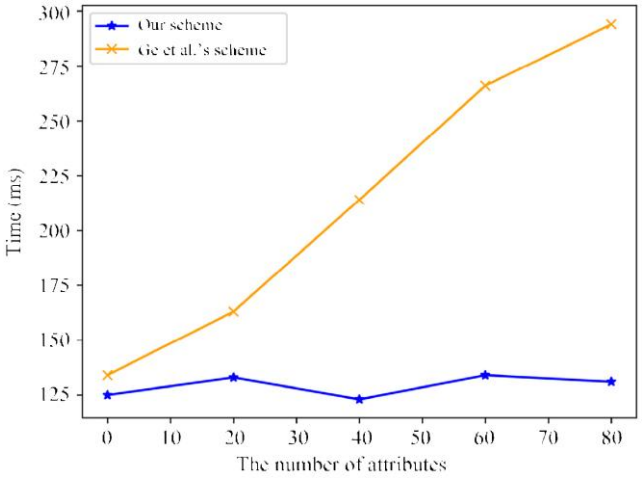

Fig. 5. Implementation time comparison of Setup algorithm.

of Setup algorithm is steady and lower than Scheme [9]. And in our scheme, in order to execute key generation, encryption and decryption algorithms successfully, we use $H_1(x)$ to replace h_i, where $x \in S$. Furthermore, scheme [9] do not support access policy hidden, which may leak some privacy information about system users. To address this problem, we outsource the attribute name to cloud servers publicly and hide the attribute value to protect privacy information. The attribute name can help users to evaluate whether their attributes are in the access policy. And the attribute value is used for key generation and data encryption.

In addition, there are some other works on ABE and SE. However, those schemes are not suitable for the power grid system because of some drawbacks. Zhang et al. [17] proposed an efficient policy-hiding attribute-based access control scheme. This scheme can support large universe and access policies partially hidden. However, in this scheme, because the algorithms are constructed based on composite order bilinear mapping, the scheme has low efficiency. Yang et al. [18] proposed a big data access control scheme and designed an attribute bloom filter to hide the whole attribute. However, the above schemes cannot support keyword search. Li et al.'s scheme [23] is based on lattice cryptography which can make the scheme against quantum attacks. However, this scheme is based on KP-ABE, which affects the practicability of the scheme. In this scheme, data owners perform key generation algorithm and generate secret keys based on their access policies. On the contrary, in the CP-ABE scheme, data users can generate secret keys based on their attributes. Therefore, using CP-ABE to achieve access control is a more reasonable approach. In addition, the access structure in this scheme can only support limited access policies. The scheme [29] can support single keyword-based searchable encryption and fine-grained access control. However, this scheme is based on "AND gate" structure and cannot support large universe. Similarly, there also are some schemes [30–32] that are based on the "AND gate on Multivalued Attributes" access structure, which limits the expressivity and flexibility of the access policy. Table 1 summarizes that our scheme supports large universe and the hidden policy. Furthermore, our scheme is based on LSSS structure that can support expressive access policies. And we use CP-ABE to achieve fine-grained access control, which makes our scheme more practical.

7. Conclusion

In this paper, we proposed an attribute-based keyword search encryption scheme for power data protection. This scheme supports both keyword-based search over encrypted data and fine-grained access control. Data owners can encrypt their power data with access policies and keywords. Data users can obtain the search trapdoor from attribute authority by using their attributes

Table 1
Comparisons of schemes.

Schemes	Keyword search	Large universe	Hidden policy	Expressiveness	Access policy
[17]	✕	✓	✓	LSSS	Ciphertext policy
[9]	✓	✕	✕	LSSS	Ciphertext policy
[23]	✓	✓	✕	AND	Key policy
[29]	✓	✕	✓	AND	Ciphertext policy
Ours	✓	✓	✓	LSSS	Ciphertext policy

and search keywords. By using the search trapdoor, cloud servers can execute search operations on behalf of data users. Only users whose attributes satisfy the access policies can search and decrypt the ciphertext. Using this scheme can achieve keyword search over the encrypted data and do not leak any privacy information of users.

In addition, because there are great varieties of users in the power system, the attribute space of the system should be large. To satisfy the requirement of the power system, we improved Ge et al.'s scheme [9]. Compared to this scheme, our scheme can support large universe requirement and the partially hidden access policy. We make the size of public parameters constant rather than increasing linearly with the number of attributes. And we hide the attribute value, which protects the privacy of user's attributes. Furthermore, because of using the LSSS structure, our proposed scheme supports expressive access policies.

Finally, the security analysis and performance analysis indicate that our scheme is efficient and practical. And the comparisons with other schemes demonstrate the advantages of our proposed scheme. Our scheme can support keyword search, large universe, hidden policy and expressive access policy. In a word, our scheme can meet the demand of the power system and achieve attribute-based keyword search for power data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Science Foundation of China (62272389), the Shenzhen Fundamental Research Program (20210317191843003), Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX2022065), Gansu Science and Technology Association Young Science and Technology Talents Lifting Project (GXH20220530-10).

References

[1] D. Cai, H. Tian, Y. Wang, H. Wang, H. Zheng, K. Cao, C. Zhou, Electric power big data and its applications, in: Proceedings of the 2016 International Conference on Energy, Power and Electrical Engineering, Atlantis Press, 2016/10, pp. 181–184, http://dx.doi.org/10.2991/epee-16.2016.39.

[2] P. Xiaosheng, D. Diyuan, C. Shijie, W. Jinyu, L. Zhaohui, N. Lin, Key technologies of electric power big data and its application prospects in smart grid, in: Proceedings of the Chinese Society of Electrical Engineering, Vol. 35, 2015, pp. 503–511.

[3] P. Mell, T. Grance, The NIST definition of cloud computing, Commun. ACM 53 (2010) 50.

[4] R.X. Lu, H. Zhu, X.M. Liu, J.K. Liu, J. Shao, Toward efficient and privacy-preserving computing in big data era, IEEE Netw. 28 (2014) 46–50, http://dx.doi.org/10.1109/MNET.2014.6863131.

[5] D.X. Song, D.A. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, 2000, pp. 44–55.

[6] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT '05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 457–473.

[7] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 89–98, http://dx.doi.org/10.1145/1180405.1180418.

[8] Q. Zheng, S. Xu, G. Ateniese, VABKS: Verifiable attribute-based keyword search over outsourced encrypted data, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 522–530.

[9] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, F. Liming, Secure keyword search and data sharing mechanism for cloud computing, IEEE Trans. Depend. Secure Comput. 18 (2021) 2787–2800, http://dx.doi.org/10.1109/TDSC.2020.2963978.

[10] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 2007 IEEE Symposium on Security and Privacy (SP '07), 2007, pp. 321–334, http://dx.doi.org/10.1109/SP.2007.11.

[11] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography, PKC '11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 53–70.

[12] A. Lewko, B. Waters, Decentralizing Attribute-Based Encryption, Springer, Berlin, Heidelberg, 2011.

[13] Z. Liu, Z. Cao, D.S. Wong, Efficient generation of linear secret sharing scheme matrices from threshold access trees, Cryptol. EPrint Arch. (2010).

[14] J. Lai, R.H. Deng, Y. Li, Expressive CP-ABE with partially hidden access structures, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 18–19, http://dx.doi.org/10.1145/2414456.2414465.

[15] T. Nishide, K. Yoneyama, K. Ohta, Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures, Springer-Verlag, 2008.

[16] L. Jin, K. Ren, Z. Bo, Z. Wan, Privacy-aware attribute-based encryption with user accountability, in: International Conference on Information Security, 2009.

[17] Y.H. 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, http://dx.doi.org/10.1109/JIOT.2018.2825289.

[18] K. Yang, Q. Han, H. Li, Z. Kan, S. Zhou, X.M. Shen, An efficient and fine-grained big data access control scheme with privacy-preserving policy, IEEE Internet Things J. 4 (2017) 563–571, http://dx.doi.org/10.1109/JIOT.2016.2571718.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: C. Cachin, J.L. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 506–522.

[20] D.J. Park, K. Kim, P.J. Lee, Public key encryption with conjunctive field keyword search, in: C.H. Lim, M. Yung (Eds.), Information Security Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 73–86.

[21] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou, Fuzzy keyword search over encrypted data in cloud computing, in: 2010 Proceedings IEEE INFOCOM, 2010, pp. 1–5, http://dx.doi.org/10.1109/INFCOM.2010.5462196.

[22] J. Li, L. Zhang, Attribute-based keyword search and data access control in cloud, in: 2014 Tenth International Conference on Computational Intelligence and Security, 2014, pp. 382–386, http://dx.doi.org/10.1109/CIS.2014.113.

[23] C. Li, M. Dong, J. Li, G. Xu, X.-B. Chen, W. Liu, K. Ota, Efficient medical big data management with keyword-searchable encryption in healthchain, IEEE Syst. J. (2022) 1–12, http://dx.doi.org/10.1109/JSYST.2022.3173538.

[24] G. Xu, Y. Cao, S. Xu, X. Liu, X.-B. Chen, Y. Yu, X. Wang, A searchable encryption scheme based on lattice for log systems in blockchain, CMC-Comput. Mater. Contin. 72 (3) (2022) 5429–5441.

[25] Y.B. Miao, X.M. Liu, K.K.R. Choo, R.H. Deng, J.G. Li, H.W. Li, J.F. Ma, Privacy-preserving attribute-based keyword search in shared multi-owner setting, IEEE Trans. Depend. Secure Comput. 18 (2021) 1080–1094, http://dx.doi.org/10.1109/TDSC.2019.2897675.

[26] A. Beimel, Secure Schemes for Secret Sharing and Key Distribution (Ph.D. Thesis), Israel Institute of Technology Technion, 1996.
[27] A.D. Caro, V. Iovino, jPBC: Java pairing based cryptography, Comput. Commun. (2011).
[28] S. Qiu, J. Liu, Y. Shi, R. Zhang, C.I. Technology, B.J. University, S.K.L. of Information Security, I. Engineering, C. Sciences, Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack, Sci. China(Inf. Sci.) (2017).
[29] P. Chaudhari, M.L. Das, Privacy preserving searchable encryption with fine-grained access control, IEEE Trans. Cloud Comput. 9 (2021) 753–762, http://dx.doi.org/10.1109/TCC.2019.2892116.
[30] L.C. Cao, J.B. Zhang, X.Y. Dong, C.Z. Xi, Y.F. Wang, Y.Y. Zhang, X. Guo, T. Feng, A based on blinded CP-ABE searchable encryption cloud storage service scheme, Int. J. Commun. Syst. 31 (2018) http://dx.doi.org/10.1002/dac.3566.
[31] A. Wu, D. Zheng, Y.H. Zhang, M.L. Yang, Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing, Sensors 18 (2018) http://dx.doi.org/10.3390/s18072158.
[32] Y.B. Miao, X.M. Liu, K.K.R. Choo, R.H. Deng, H.J. Wu, H.W. Li, Fair and dynamic data sharing framework in cloud-assisted internet of everything, IEEE Internet Things J. 6 (2019) 7201–7212, http://dx.doi.org/10.1109/JIOT.2019.2915123.